DB2

IBM

DB2 版本 9

适用于 **Linux**、**UNIX** 和 **Windows**

故障诊断指南

G151-0285-00

DB2

IBM

DB2 版本 9
适用于 **Linux**、**UNIX** 和 **Windows**

故障诊断指南

G151-0285-00

在使用本资料及其支持的产品之前，请务必阅读『*声明*』中的一般信息。

版本声明

本文档包含 IBM 的专利信息。它是根据许可协议提供的，并受版权法保护。本出版物包含的信息不包括任何产品保证，且本手册提供的任何声明不应作如此解释。

可以用在线方式或通过您当地的 IBM 代表订购 IBM 出版物。

- 要以在线方式订购出版物，可访问 IBM 出版物中心（IBM Publications Center），网址为 www.ibm.com/shop/publications/order。
- 要查找您当地的 IBM 代表，可访问 IBM 全球联系人目录（IBM Directory of Worldwide Contacts），网址为 www.ibm.com/planetwide。

在美国或加拿大，要从 “DB2 市场营销和销售中心” 订购 DB2 出版物，请致电 1-800-IBM-4YOU（426-4968）。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

© Copyright International Business Machines Corporation 1993, 2006. All rights reserved.

目录

第 1 章 了解更多信息 1
问题确定简介 1
关于首次故障数据捕获（FFDC） 3
首次故障数据捕获信息 3
一般首次故障数据捕获（FFDC） 3
关于管理通知日志文件 5
管理通知日志 5
设置管理通知日志文件的错误捕获级别 5
解释管理通知日志文件条目 5
关于 DB2 诊断日志（db2diag.log）文件 7
设置诊断日志文件错误捕获级别 7
解释 db2diag.log 文件参考记录 7
解释诊断日志文件条目 8
db2cos（调出脚本）输出文件 10
转储文件 12
关于陷阱文件 13
陷阱文件 13
格式化陷阱文件（Windows） 13
关于特定于平台的错误日志 14
特定于平台的错误日志信息 14
系统核心文件（UNIX） 15
访问系统核心文件信息（UNIX） 15
访问事件日志（Windows） 16
导出事件日志（Windows） 17
访问 Dr. Watson 日志文件（Windows） 17
组合 DB2 数据库和操作系统诊断 17

第 2 章 DB2 故障诊断 21
DB2 故障诊断 21
分区数据库环境故障诊断 21
在分区数据库环境中发出命令 21

第 3 章 DB2 Connect 故障诊断 23
问题确定 23
收集相关信息 23
初始连接不成功 24
初始连接之后遇到的问题 24
诊断工具 25

第 4 章 用于故障诊断的工具 27
db2dart 工具概述 27
使用 db2diag 分析 db2diag.log 文件 27
使用 db2greg 显示和改变全局注册表（UNIX） . . . 28
标识产品的版本和服务级别 29
使用 db2look 模拟数据库 29
列示安装在系统上的 DB2 产品（Linux 和 UNIX） 32
使用 db2pd 进行监视和故障诊断 34
使用 db2support 收集环境信息 42
跟踪 44
基本跟踪诊断 44
DB2 跟踪 45
DRDA 跟踪 48
控制中心跟踪 57
JDBC 跟踪文件 58
CLI 跟踪 61
特定于平台的工具 66
诊断工具（Windows） 66
诊断工具（Linux 和 UNIX） 67

第 5 章 搜索知识库 69
如何有效地搜索已知问题 69
问题确定资源 69

第 6 章 获取修订 71
应用修订包 71

附录 A. DB2 数据库技术信息 73
DB2 技术信息概述 73
文档反馈 73
PDF 格式的 DB2 技术资料库 74
订购印刷版 DB2 书籍 76
从命令行处理器显示 SQL 状态帮助 77
访问不同版本的 DB2 信息中心 77
以首选语言显示 DB2 信息中心中的主题 77
更新安装在计算机或内部网服务器上的 DB2 信息中心 78
DB2 教程 80
DB2 故障诊断信息 80
条款和条件 81

附录 B. 声明 83
商标 84

索引 87

与 IBM 联系 91

© Copyright IBM Corp. 1993, 2006

第 1 章 了解更多信息

问题确定简介

为了准确地分析问题，第一步要做的就是完整地描述问题。如果没有问题描述，您就不知道从什么地方开始调查造成问题的原因。此步骤包括询问自己如下基本问题:

- 症状是什么?
- 问题是在哪里发生的?
- 问题是在何时发生的?
- 发生问题的条件是什么?
- 问题是否可以再现?

通过回答上述及其他问题，就得到了对大多数问题的准确描述，并且也是找出问题解决方案的最好办法。

症状是什么?:

开始描述问题时，最明显的问题是“发生了什么问题？”。这看起来像一个很直观的问题；但是，它可以分为若干其他问题，从而更好地描述该问题。这些问题包括:

- 是什么人或什么工具报告该问题的?
- 错误代码和错误消息是什么?
- 怎么失败的？例如: 循环、挂起、崩溃、性能下降或结果错误。
- 对企业有什么影响?

问题是在哪里发生的?:

确定问题发生的位置并不总是那么容易，但它是解决问题的最重要步骤之一。报告组件和失败组件之间可能存在多层技术。网络、磁盘和驱动程序只是调查问题时要考虑的几个部分。

- 是在特定平台上还是在多个平台上都发生了该问题?
- 是否支持当前环境和配置?
- 应用程序是在数据库服务器本地运行还是在远程服务器上运行?
- 是否涉及网关?
- 数据库是驻留在各个磁盘上，还是驻留在 RAID 磁盘阵列上?

这些类型的问题可帮助您隔离问题层，并且是确定问题来源所必需的。记住，不能只因为某层报告问题而总是断定那就是问题根源所在。

标识发生问题的位置时应了解发生问题的环境。总是应该花一些时间来完整描述问题环境，包括操作系统、操作系统版本、所有相应软件及版本和硬件信息。确认您正在配置受支持的环境中运行，这是因为许多问题会解释为发现若干软件级别不能在一起运行，或者未在一起经过完整测试。

问题是在何时发生的?:

© Copyright IBM Corp. 1993, 2006

问题分析中的另一个必需步骤是创建导致故障的事件的详细时间线，对于从前发生的那些案例尤其如此。可以通过回溯工作过程很轻松地完成此任务：从报告错误时开始（尽可能精确，甚至精确到毫秒），然后通过可用日志和信息回溯工作过程。通常您只需要进行观察，直到您在任何诊断日志中发现的第一个值得怀疑的事件，但是，这并不总是那么容易，通常需要您有实践经验。如果同时存在多层技术，每层都有自己的诊断信息，则要知道停止的时间就特别困难。

- 问题是否仅在日间或夜间的特定时间发生？
- 问题多久发生一次？
- 导致问题的事件的发生顺序是什么？
- 问题是否发生在环境改变（如升级现有软件或硬件或者安装新的软件或硬件）之后？

回答这类问题可帮助您创建事件的详细时间线，并且为您提供用来进行调查的参考框架。

发生问题的条件是什么？：

要完整地描述问题，知道发生问题时还有什么别的软件在运行是非常重要的。如果问题是在特定环境或特定条件下发生的，则这可能是找出问题原因的关键线索。

- 问题是否总是在执行同一任务时发生？
- 事件是否要按一定顺序发生，问题才会再现？
- 是否存在其他应用程序同一时间失败？

回答这些类型的问题可帮助您说明发生问题的环境，并且能够将所有从属项关联起来。记住，不能只因为同一时间发生了多个问题就表示它们总是相关的。

问题是否可以再现？：

从问题描述和调查角度来看，“理想”的问题是可以再现的。对于可再现的问题，您几乎总是可以将它们与提供的一大堆工具或过程配合使用，以帮助进行调查。因此，可再现问题通常比较容易调试和解决。

但是，可再现问题也有缺点：如果该问题对企业有很严重的影响，则您可能不想让它再现。在这种情况下，最好尽可能在测试或开发环境中再现该问题。

- 能否在测试机器上再现问题？
- 是否多个用户或应用程序都遇到同一类型的问题？
- 能否通过运行单个命令、一组命令、特定应用程序或独立应用程序来再现问题？
- 能否通过从 DB2® 命令行输入等价命令 / 查询来再现问题？

因为在调查时测试或开发环境有更好的灵活性，也更便于控制，所以最好在测试或开发环境中再现容易发生的单个问题。

相关概念:

- 第 69 页的『问题确定资源』

关于首次故障数据捕获（FFDC）

首次故障数据捕获信息

首次故障数据捕获（FFDC）是一个术语，适用于发生错误时 DB2 数据库产品自动捕获的诊断信息集合。此信息减少了再现错误以获取诊断信息的需要。

有关特定类型的 FFDC 的更多详细信息，请参阅下面列示的相关链接。

相关概念:

- 第 12 页的『转储文件』
- 第 13 页的『陷阱文件』

相关参考:

- 第 3 页的『一般首次故障数据捕获（FFDC）』

一般首次故障数据捕获（FFDC）

管理通知日志（“*instance_name*.nfy”）

- 操作系统：所有操作系统
- 缺省位置:
  - Linux™ 和 UNIX®：位于 *diagpath* 数据库管理器配置参数所指定的目录中。
  - Windows®：使用事件查看器工具（“开始”>“控制面板”>“管理工具”>“事件查看器”）
- 创建实例时自动创建。
- 发生重大事件时，DB2 将信息写入管理通知日志。该信息供数据库和系统管理员使用。记录在此文件中的消息类型由 *notifylevel* 配置参数确定。

DB2 诊断日志（“db2diag.log”）

- 操作系统：所有操作系统
- 缺省位置：位于 *diagpath* 数据库管理器配置参数所标识的目录中。
- 创建实例时自动创建。
- 此文本文件包含关于实例遇到的错误和警告的诊断信息。此信息用于问题确定及 IBM® 客户支持。记录在此文件中的消息类型由 *diaglevel* 数据库管理器配置参数确定。

DB2 管理服务器（DAS）诊断日志（“db2dasdiag.log”）

- 操作系统：所有操作系统
- 缺省位置:
  - Linux 和 UNIX：位于 DASHOME/das/dump 中，其中 DASHOME 是 DAS 所有者的主目录
  - Windows：位于 DAS 主目录的“dump”文件夹中。例如，C:\Program Files\IBM\SQLLIB\DB2DAS00\dump
- 创建 DAS 时自动创建。
- 此文本文件包含关于 DAS 遇到的错误和警告的诊断信息。

DB2 事件日志（“db2eventlog.xxx”，其中 xxx 是数据库分区号）

- 操作系统: 所有操作系统
- 缺省位置: 位于 *diagpath* 数据库管理器配置参数所指定的目录中。
- 创建实例时自动创建。
- DB2 事件日志文件是数据库管理器中发生的基础结构级事件的循环日志。该文件大小固定，并且充当在实例运行时记录的特定事件的循环缓冲区。每次停止实例时，就会替换先前的事件日志，而不是追加。如果实例捕获，则还会生成 db2eventlog.XXX.crash 文件。这些文件供 IBM 客户支持使用。

DB2 调出脚本（db2cos）输出文件

- 操作系统: 所有操作系统
- 缺省位置: 位于 *diagpath* 数据库管理器配置参数所指定的目录中。
- 出现应急启动、陷阱或分段违例时自动创建。还可以在使用 **db2pdcfg** 命令所指定的特定问题情况期间创建。
- 缺省 db2cos 脚本将调用 **db2pd** 命令以打开方式收集信息。根据 db2cos 脚本中包含的命令，db2cos 输出文件的内容会有所不同。

转储文件

- 操作系统: 所有操作系统
- 缺省位置: 位于 *diagpath* 数据库管理器配置参数所指定的目录中。
- 出现特定问题情况时自动创建。
- 对于某些错误情况，会将附加信息记录在以失败进程标识命名的二进制文件中。这些文件供 IBM 客户支持使用。

陷阱文件

- 操作系统: 所有操作系统
- 缺省位置: 位于 *diagpath* 数据库管理器配置参数所指定的目录中。
- 实例异常结束时自动创建。还可以使用 **db2pd** 命令创建。
- 如果数据库管理器由于陷阱、分段违例或异常而不能继续处理，则会生成陷阱文件。

核心文件

- 操作系统: Linux 和 UNIX
- 缺省位置: 位于 *diagpath* 数据库管理器配置参数所指定的目录中。
- DB2 实例异常终止时由操作系统创建。
- 核心文件是一个二进制文件，它包含类似于 DB2 数据库产品生成的陷阱文件的信息。核心文件还可能包含已终止进程的完整内存映像。

相关概念:

- 第 5 页的『管理通知日志』
- 第 10 页的『db2cos（调出脚本）输出文件』
- 第 12 页的『转储文件』
- 第 5 页的『解释管理通知日志文件条目』
- 第 13 页的『陷阱文件』

关于管理通知日志文件

管理通知日志

发生重大事件时，DB2 数据库管理器将信息写至管理通知日志。例如，它记录 DB2 实用程序（REORG、BACKUP、RECOVERY 和 ROLL-FORWARD）的状态、高级别应用程序问题、许可证发放活动、日志文件路径和存储问题、监视和建立索引活动以及表空间问题等。此信息是供数据库和系统管理员使用的。

通知消息提供了其他信息以补充提供的 SQLCODE。事件的类型和所收集信息的详细级别是由 NOTIFYLEVEL 配置参数确定的。详细的诊断信息记录在 db2diag.log 文本日志文件中，而不在管理日志中。诊断信息用于问题确定以及 DB2 客户支持。详细信息级别由 DIAGLEVEL 配置参数确定。

相关概念:

- 第 5 页的『解释管理通知日志文件条目』
- 第 8 页的『解释诊断日志文件条目』
- 第 7 页的『解释 db2diag.log 文件参考记录』

相关任务:

- 第 7 页的『设置诊断日志文件错误捕获级别』
- 第 5 页的『设置管理通知日志文件的错误捕获级别』

相关参考:

- 『notifylevel - 通知级别配置参数』（*《性能指南》*）
- 『diaglevel - 诊断错误捕获级别配置参数』（*《性能指南》*）

设置管理通知日志文件的错误捕获级别

DB2 记录在管理通知日志中的信息由 NOTIFYLEVEL 设置确定。要检查当前设置，发出:

```
DB2 GET DBM CFG
```

查找以下变量:

```
通知级别（NOTIFYLEVEL）                    = 3
```

要改变该设置，使用以下命令:

```
DB2 UPDATE DBM CFG USING NOTIFYLEVEL X
```

其中 X 是期望的通知级别。

相关参考:

- 『notifylevel - 通知级别配置参数』（*《性能指南》*）

解释管理通知日志文件条目

使用文本编辑器来查看怀疑发生了问题的机器上的管理通知日志文件。记录的最新事件在文件的最后面。通常，每个条目包含下列部分:

- 时间戳记
- 报告错误的位置。应用程序标识允许您匹配在服务器和客户机的日志上与应用程序有关的条目。
- 说明错误的诊断消息（通常以“DIA”或“ADM”开头）。
- 任何可用的支持数据，例如 SQLCA 数据结构和指向任何其他转储文件或陷阱文件的位置的指针。

如果数据库行为正常，则此类信息不是很重要且可以忽略。

“管理”日志是不断增长的。当它们变得太大时，将它们备份然后擦除文件。当下一次系统需要它们时，会自动生成一组新的文件。

以下示例显示样本日志条目的标题信息，且标识了日志的所有部分。

注: 不是所有日志条目都包括所有这些部分。

```
2006-02-15-19.33.37.630000 1     实例: DB2 2     节点: 000 3
PID: 940 (db2syscs.exe) TID: 660 4     应用程序标识: *LOCAL.DB2.020205091435 5
恢复管理器 6   sqlpresr 7   探测: 1 8     数据库: SAMPLE   9
ADM1530E 10   已启动崩溃恢复。 11
```

图注:

1. 消息的时间戳记。
2. 生成该消息的实例的名称。
3. 对于多分区系统，生成消息的分区。（在非分区数据库中，该值为“000”。）
4. 写入消息的 DB2 组件。对于由用户应用程序使用 `db2AdminMsgWrite` API 写入的消息，该组件将为“用户应用程序”。
5. 进程正在为其工作的应用程序的标识。在本示例中，生成消息的进程代表标识为 *LOCAL.DB2.020205091435 的应用程序工作。

   要标识关于特定应用程序标识的信息，执行下列其中一项操作:

   - 在 DB2 服务器上使用 **db2 list applications** 命令或在 DB2 Connect™ 网关上使用 **db2 list dcs applications** 来查看应用程序标识列表。可以根据此列表确定有关遇到错误的客户机的信息，例如其节点名以及其 TCP/IP 地址。
   - 使用 **db2 get snapshot for application** 命令来查看应用程序标识列表。
6. 写入消息的 DB2 组件。对于由用户应用程序使用 `db2AdminMsgWrite` API 写入的消息，该组件将为“用户应用程序”。
7. 提供消息的函数的名称。此函数在写入消息的 DB2 子组件中运行。对于由用户应用程序使用 `db2AdminMsgWrite` API 写入的消息，该函数将为“用户函数”。

   要查找有关函数执行的活动的类型的更多信息，查看其名称的第四个字母。在本示例中，函数“sqlpresr”中的第四个字母“p”指示数据保护问题。（例如，日志可能毁坏。）

   以下列表显示函数名的第四位使用的一些字母以及它们标识的活动类型:

   b 缓冲池

   c 客户机与服务器之间的通信

   d 数据管理

   e 引擎进程

o 操作系统调用（例如打开和关闭文件）

p 数据保护（例如锁定和记录）

r 关系数据库服务

s 排序

x 建立索引

8. 唯一内部标识。此编号允许 DB2 客户支持和开发团队在 DB2 源代码中找到报告该消息的位置。
9. 发生错误的数据库。
10. 以十六进制代码指示错误类型和编号的消息（如果有的话）。
11. 说明记录的事件的消息文本（如果有的话）。

相关概念:

- 第 5 页的『管理通知日志』
- 第 3 页的『首次故障数据捕获信息』

相关参考:

- 『notifylevel - 通知级别配置参数』（《*性能指南*》）

关于 DB2 诊断日志（db2diag.log）文件

设置诊断日志文件错误捕获级别

DB2 诊断日志是包含 DB2 记录的文本信息的文件。此信息用于问题确定及 DB2 客户支持。

DB2 在 db2diag.log 中记录的信息由 DIAGLEVEL 设置确定。要检查当前设置，发出:

```
DB2 GET DBM CFG
```

查找以下变量:

```
诊断错误捕获级别（DIAGLEVEL）                = 3
```

要改变该设置，使用以下命令:

```
DB2 UPDATE DBM CFG USING DIAGLEVEL X
```

其中 X 是期望的通知级别。

注: 如果要诊断可再现的问题，则建议您在执行问题确定时使用 DIAGLEVEL 4。

相关概念:

- 第 3 页的『首次故障数据捕获信息』

相关参考:

- 『diaglevel - 诊断错误捕获级别配置参数』（《*性能指南*》）

解释 db2diag.log 文件参考记录

db2diag.log 中的第一个消息应该总是参考记录。

唯一的例外是以下情况：日志文件中的第一条消息由使用 ossLog API（如 DAS 和 genreg 等等）调用的组件生成，而这种情况下没有参考记录输出。

参考记录的示例如下所示:

```
2006-02-09-18.07.31.059000-300 I1H917              级别: 事件
PID: 3140                 TID: 2864        PROC: db2start.exe
实例: DB2                 节点: 000
函数: DB2 UDB, RAS/PD 组件, _pdlogInt, 探测: 120
开始: 新的诊断日志文件
DATA #1: 构建级别, 124 字节
实例 "DB2" 使用 "32" 位和 DB2 代码发行版 "SQL09010", 其级别标识为 "01010107"。
参考标记为 "DB2 v9.1.0.190"、 "s060121"、 "" 和修订包 "0"。
DATA #2 : 系统信息, 1564 字节
系统: WIN32_NT MYSRVR Service Pack 2 5.1 x86 系列 15, 模型 2, 步骤 4
CPU: 总计: 1 联机: 1 每个套接字的核心数: 1 每个核心的线程化程度: 1
物理内存 (MB): 总计: 1024 空闲: 617 可用: 617
虚拟内存 (MB): 总计: 2462 空闲: 2830
交换内存 (MB): 总计: 1438 空闲: 2213
此记录中的信息仅在创建此文件时有效 (请参阅此记录的时间戳记)
```

参考记录是 "db2start" 在每个逻辑分区上的输出。这会生成多个参考记录：每个逻辑分区一个参考记录。因为参考记录包含的内存值在每个分区上都不同，所以此信息可能很有用。

解释诊断日志文件条目

使用文本编辑器来查看怀疑发生了问题的机器上的诊断日志文件。记录的最新事件在文件的最后面。

注: "管理"日志是*不断*增长的。当它们变得太大时，将它们备份然后擦除文件。当下一次系统需要它们时，会自动生成一组新的文件。

以下示例显示样本日志条目的标题信息，且标识了日志的所有部分。

注: 不是所有日志条目都包括所有这些部分。只有开头的一些字段（时间戳记至 TID）和 FUNCTION 才会显示在所有 db2diag.log 记录中。

```
2006-02-13-14.34.35.965000-300 1    I17502H435 2        级别: 错误 3
PID: 940 4                 TID: 660 5     PROC: db2syscs.exe 6
实例: DB2 7                 节点: 000 8      数据库: SAMPLE 9
APPHDL: 0-1433 10                APPID: *LOCAL.DB2.050120082811 11
FUNCTION: 12 DB2 UDB, 数据保护, sqlpsize, 探测: 20
RETCODE: 13 ZRC=0x860F000A=-2045837302=SQLO_FNEX "找不到文件。"
          DIA8411C 找不到文件 ""。
```

图注:

1. 消息的时间戳记和时区。
2. 记录标识字段。对于创建 DB2 诊断日志的平台，db2diag.log 的记录标识指定要记录的当前消息的文件位移（如 "17502" ）和消息长度（如 "435" ）。
3. 与错误消息相关联的诊断级别。例如，信息、警告、错误、严重或事件。
4. 进程标识。
5. 线程标识
6. 进程名称
7. 生成该消息的实例的名称。

8. 对于多分区系统，此项为生成该消息的分区。（在非分区数据库中，该值为“000”。）

9. 数据库名称

10. 应用程序句柄。此值与 db2pd 输出和锁定转储文件中使用的值相匹配。它包括后跟协调程序索引编号并且用破折号分开的协调程序分区号。

11. 进程正在为其工作的应用程序的标识。在本示例中，生成消息的进程代表标识为 *LOCAL.DB2.050120082811 的应用程序工作。

    TCP/IP 生成的应用程序标识由三个部分组成

    1. **IP 地址**：它表示为 32-bit 位数字，最大显示为 8 位十六进制字符。
    2. **端口号**：显示为 4 位十六进制字符。
    3. 此应用程序的实例的**唯一标识**。

    注：当 IP 地址或端口号的十六进制版本以 0 至9 开头，则它们将分别转换为 G 至 P。例如，“0”映射为“G”，“1”映射为“H”，以此类推。IP 地址 AC10150C.NA04.006D07064947 表示为如下所示：IP 地址仍为 AC10150C，它将转换为 172.16.21.12。端口号为 NA04。第一个字符为“N”，它将映射为“7”。因此，端口号的十六进制为 7A04，它将转换为十进制格式 31236。

    要标识关于特定应用程序标识的信息，执行下列其中一项操作:

    - 在 DB2 服务器上使用 **db2 list applications** 命令或在 DB2 Connect 网关上使用 **db2 list dcs applications** 来查看应用程序标识列表。可以根据此列表确定有关遇到错误的客户机的信息，例如其节点名以及其 TCP/IP 地址。
    - 使用 **db2 get snapshot for application** 命令来查看应用程序标识列表。
    - 使用 **db2pd -applications -db sample** 命令。

12. 写入消息的产品名（“DB2”）、组件名称（“数据保护”）和函数名（“sqlpsize”）以及函数中的探测点（“20”）。

13. 被调用函数返回的返回码（如果有的话）。此字段由类型（“ZRC”）、返回码值和相应的错误描述组成。

注：db2diag.log 中的时间戳记包含时区。例如：2006-02-13-14.34.35.965000-300，其中“-300”是 UTC（全球标准时间，以前称为 GMT）与应用程序服务器上的当地时间之差（**以分钟计**）。因此，-300 表示 UTC - 5 小时，如 EST（东部标准时间）。

既然已经看到了样本 db2diag.log 条目，以下是所有可能字段的列表:

```
<timestamp><timezone>       <recordID>          级别: <level> (<source>)
PID: <pid>              TID: <tid>       PROC: <procName>
实例: <instance>        节点: <node>     数据库: <database>
APPHDL: <appHandle>     APPID: <appID>
函数: <prodName>, <compName>, <funcName>, 探测: <probeNum>
消息: <messageID>  <msgText>
已调用: <prodName>, <compName>, <funcName>   OSERR: <errorName> (<errno>)
RETCODE: <type>=<retCode> <errorDesc>
ARG #N: <typeTitle>, <typeName>, <size> bytes
... argument ...
DATA #N: <typeTitle>, <typeName>, <size> bytes
... data ...
```

尚未在示例中说明的字段包括:

- <source> 指示已记录错误的源头。可能的值包括:
  - origin - 导致错误的函数（起始点）已记录消息
  - OS - 操作系统已生成错误
  - received - 另一进程（客户机／服务器）已接收到错误
  - sent - 错误已发送至另一进程（客户机／服务器）
- MESSAGE 包含要记录的消息。它包括:
  - <messageID> - 消息号，如 ECF=0x9000004A 或 DIA8604C
  - <msgText> - 错误描述

  如果还有被调用（CALLED）字段，则 <msgText> 是被调用（CALLED）函数返回的错误对记录消息的函数（在 FUNCTION 字段中指定）的影响
- 被调用（CALLED） 这是返回错误的函数。它包括:
  - <prodName> - 产品名: “OS”、“DB2”、“DB2 Tools”或“DB2 Common”
  - <compName> - 组件名称（如果是系统调用则为“-”）
  - <funcName> - 被调用函数名
- OSERR 这是被调用（CALLED）的系统调用返回的操作系统错误。它包括:
  - <errorName> - 特定于系统的错误名称
  - <errno> - 操作系统错误号
- ARG 本节列示返回错误的函数调用的自变量。它包括:
  - <N> - 对“被调用”函数的调用中的自变量的位置
  - <typeTitle> - 与第 N 个自变量类型名相关联的标注
  - <typeName> - 要记录的自变量的类型的名称
  - <size> - 要记录的自变量的大小
- DATA 它包含记录函数转储的所有其他数据。它包括:
  - <N> - 要转储的数据对象的序号
  - <typeTitle> - 要转储的数据的标注
  - <typeName> - 要记录的数据字段的类型的名称，如 PD_TYPE_UINT32 和 PD_TYPE_STRING
  - <size> - 数据对象的大小

相关概念:

- 第 7 页的『解释 db2diag.log 文件参考记录』

db2cos（调出脚本）输出文件

在缺省情况下，当数据库管理器因为应急启动、陷阱、分段违例或异常而不能继续处理时，将会调用 db2cos 脚本。缺省 db2cos 脚本将调用 db2pd 命令以打开方式收集信息。

在多分区配置中，将仅对遇到陷阱的分区上的陷阱代理程序调用该脚本。如果需要从其他分区收集信息，可更新 db2cos 脚本以使用 **db2_all**命令，或者如果所有分区在同一台机器上，则在 **db2pd** 命令上指定 **-alldbpartitionnums** 选项。

还可通过 **db2pdcfg -cos** 命令来配置触发 db2cos 调用的信号类型。缺省配置用于要在发生应急启动或陷阱时运行的 db2cos 脚本。另一方面，在缺省情况下，生成的信号不会启动 db2cos 脚本。

发生应急启动、陷阱、分段违例或异常时，事件顺序如下所示:

1. 创建陷阱文件
2. 调用信号处理程序
3. 调用 db2cos 脚本（取决于启用的 db2cos 设置）
4. 在管理通知日志中记录相应条目
5. 在 db2diag.log 中记录相应条目

db2cos 脚本中的 **db2pd** 命令收集到的缺省信息包括有关操作系统、已安装 DB2 产品的版本和服务级别、数据库管理器和数据库配置的详细信息，以及有关以下各项的状态的信息：代理程序、内存池、内存块、应用程序、实用程序、事务、缓冲池、锁定、事务日志、表空间和容器。此外，它还会提供有关下列各项的信息：动态、静态和目录高速缓存的状态、表和索引统计信息、恢复状态以及重新优化的 SQL 语句及活动语句列表。如果需要收集进一步的信息，则只需使用附加命令更新 db2cos 脚本。

调用缺省 db2cos 脚本时，它将在 DIAGPATH 数据库管理器配置参数指定的目录中生成输出文件。这些文件名为 db2cosXXXYYY.ZZZ，其中 XXX 是进程标识（PID），YYY 是线程标识（TID），而 ZZZ 是数据库分区号（对于单分区数据库则为 000）。如果存在多线程陷阱，则会对每个线程单独调用 db2cos 脚本。如果 PID 和 TID 组合多次出现，则该数据将追加至文件。还会显示时间戳记，所以您可以区分输出的迭代。

根据 db2cos 脚本中指定的命令，db2cos 输出文件将包含不同信息。如果未改变缺省脚本，则会显示类似以下的条目（后跟详细 db2pd 输出）:

```
2005-10-14-10.56.21.523659
PID     : 782348               TID  : 1           PROC : db2cos
实例: db2inst1                 节点: 0            DB   : SAMPLE
APPHDL  :                      APPID: *LOCAL.db2inst1.051014155507
函数: 操作系统服务，sqloEDUCodeTrapHandler，探测: 999
事件: 从操作系统服务 sqloEDUCodeTrapHandler 调用
          /home/db2inst1/sqllib/bin/db2cos 捕获到陷阱
实例 db2inst1 使用 64 位和 DB2 代码发行版 SQL09010
...
操作系统信息:

操作系统名称: AIX
节点名: n1
版本: 5
发行版: 2
机器: 000966594C00

...
```

db2diag.log 还将包含与发生位置有关的条目。例如:

```
2005-10-14-10.42.17.149512-300 I19441A349         级别: 事件
PID: 782348               TID: 1           PROC: db2sysc
实例: db2inst1            节点: 000
函数: DB2 UDB，跟踪服务，pdInvokeCalloutScript，探测: 10
开始: 从操作系统服务 sqloEDUCodeTrapHandler 调用 /home/db2inst1/sqllib/bin/db2cos
2005-10-14-10.42.23.173872-300 I19791A310         级别: 事件
PID: 782348               TID: 1           PROC: db2sysc
实例: db2inst1            节点: 000
```

```
函数: DB2 UDB, 跟踪服务, pdInvokeCalloutScript, 探测: 20
停止: 完成 /home/db2inst1/sqllib/bin/db2cos 调用

2005-10-14-10.42.23.519227-300 E20102A509         级别: 严重
PID: 782348              TID: 1           PROC: db2sysc
实例: db2inst1           节点: 000
函数: DB2 UDB, 操作系统服务, sqloEDUCodeTrapHandler, 探测: 10
消息: ADM0503C 发生了意外的内部处理错误。与此实例相关联的所有 DB2 进程都将关闭。
      已记录诊断信息, 请与 IBM 支持机构联系以获取进一步的帮助。
2005-10-14-10.42.23.520111-300 E20612A642         级别: 严重
PID: 782348              TID: 1           PROC: db2sysc
实例: db2inst1           节点: 000
函数: DB2 UDB, 操作系统服务, sqloEDUCodeTrapHandler, 探测: 20
DATA #1: 接收到信号编号, 4 字节
11
DATA #2: 信号信息, 64 字节
0x0FFFFFFFFFFFD5C0 : 0000 000B 0000 0000 0000 0009 0000 0000    ................
0x0FFFFFFFFFFFD5D0 : 0000 0000 0000 0000 0000 0000 0000 0000    ................
0x0FFFFFFFFFFFD5E0 : 0000 0000 0000 0000 0000 0000 0000 0000    ................
0x0FFFFFFFFFFFD5F0 : 0000 0000 0000 0000 0000 0000 0000 0000    ................
```

相关概念:

- 第 3 页的『首次故障数据捕获信息』
- 第 13 页的『陷阱文件』

相关参考:

- 『db2pd - Monitor and troubleshoot DB2 database command』（*Command Reference*）
- 『db2pdcfg - Configure DB2 database for problem determination behavior command』（*Command Reference*）

转储文件

转储文件是在发生错误时创建的，它包含将有助于诊断问题（例如，内部控制块）的其他信息。写至转储文件的每个数据项都具有与其相关联的时间戳记，以帮助进行问题确定。转储文件使用二进制格式，目的是供 DB2 客户支持代表使用。

创建或追加转储文件时，会在 db2diag.log 中建立一个条目，指示所写数据的时间和类型。这些 db2diag.log 条目类似于以下所示:

```
2006-03-16-11.53.18.001160    实例: payroll    节点: 000
PID:44829(db2agent (SAMPLE))     Appid:*LOCAL.payroll.970317140834
关系数据服务          sqlrerlg     探测: 17    数据库: SAMPLE
DIA9999E 出现内置返回码。报告以下内容: "0xFFFFE101"。
转储文件: /home/db2/sqllib/db2dump/56772.000 数据: SECTION STMT  1
```

图注:

1 在此 UNIX 示例中，SECTION STMT 数据存储在名为 56772.000 的文件中，该文件位于 /home/db2/sqllib/db2dump 目录中。

注意:

- 对于分区数据库系统，文件扩展名标识分区号。例如，以下条目指示转储文件是由在分区 10 上运行的 DB2 进程创建的:

  ```
  转储文件: /home/db2/sqllib/db2dump/56772.010  数据: SECTION STMT
  ```

相关概念:

- 第 3 页的『首次故障数据捕获信息』

关于陷阱文件

陷阱文件

如果 DB2 由于陷阱、分段违规或异常而不能继续处理，它就会生成陷阱文件。

DB2 接收到的所有信号或异常都会记录在陷阱文件中。陷阱文件还包含发生错误时正在运行的函数序列。此序列有时又称“函数调用堆栈”或“堆栈回溯”。陷阱文件还包含有关捕获到信号或异常时进程的状态的其他信息。

文件位于由 DIAGPATH 数据库管理器配置参数指定的目录中。

在 Linux 和 UNIX 上，文件名的第一个字母是“t”，后跟进程标识（PID）。文件扩展名就是分区号（在单一分区数据库上为 000）。

在 Windows 上，每个陷阱文件名为 Pxxxxx.yyy，其中 xxxxx 是 PID，而 yyy 是数据库分区号（在单分区数据库上为 000）。如果由于异常而生成陷阱文件，则其扩展名将为 .TRP。

还有一些诊断陷阱，它们是在发生某些特定条件（这些条件不一定会使实例崩溃）时由代码生成的，在查看堆栈时非常有用。这些陷阱是使用 PID 以十六进制格式命名的，后跟分区号（在单分区数据库中为 0）。

示例:

- t56772.000 是具有 pid 56772 的进程的陷阱文件。
- t56772.010 是具有 pid 56772 的进程的陷阱文件。它是由在分区 10 上运行的 DB2 进程创建的。

可将 **db2pd** 命令与 -stack all 或 -dump 选项配合使用，以根据需要生成陷阱文件。尽管一般情况下只有在 DB2 支持机构请求时才应完成此任务。

相关概念:

- 第 3 页的『首次故障数据捕获信息』
- 第 15 页的『系统核心文件（UNIX）』

相关任务:

- 第 34 页的『使用 db2pd 进行监视和故障诊断』

相关参考:

- 『db2pd - Monitor and troubleshoot DB2 database command』（*Command Reference*）

格式化陷阱文件（Windows）

一个称为 db2xprt.exe 的工具，它允许您格式化陷阱文件（*.TRP）。它会将 DB2 数据库二进制陷阱文件格式化为人们可以阅读的 ASCII 文件。

该工具使用 DB2 符号文件来格式化陷阱文件。这些 .PDB 文件的子集与 DB2 数据库产品包括在一起。

如果在 DIAGPATH 中生成了陷阱文件 “DB30882416.TRP”，则可以按如下方式进行格式化:

```
db2xprt DB30882416.TRP DB30882416.FMT
```

相关概念:

- 第 3 页的『首次故障数据捕获信息』
- 第 13 页的『陷阱文件』

相关参考:

- 『db2xprt - Format trap file command』（*Command Reference*）

关于特定于平台的错误日志

特定于平台的错误日志信息

在 DB2 之外也提供了许多其他文件和实用程序，可用于帮助分析问题。通常，在确定问题的根本原因时，它们和提供的 DB2 文件一样重要。在下列领域的日志和跟踪中包含了部分这种信息:

- 操作系统
- 应用程序和第三方供应商
- 硬件

除此处描述的一些领域之外，还有许多其他领域包含这种信息，这取决于您的操作环境。因此，在调试系统中的问题时，要了解需要调查的所有潜在领域。

操作系统:

每个操作系统都有一组自己的诊断文件，用于跟踪活动和故障。最常见的（通常也是最有用的）诊断文件是错误报告或事件日志。以下是收集此信息的方法列表:

- AIX®: 使用 **/usr/bin/errpt -a** 命令
- Solaris: 使用 /var/adm/messages* 文件或 **/usr/bin/dmesg** 命令
- Linux: 使用 /var/log/messages* 文件或 **/bin/dmesg** 命令
- HP-UX: 使用 /var/adm/syslog/syslog.log 文件或 **/usr/bin/dmesg** 命令
- Windows: 使用系统日志文件、安全性日志文件、应用程序事件日志文件以及 windir\drwtsn32.log 文件（其中，windir 是 Windows 安装目录）

每个操作系统总是有更多的跟踪和调试实用程序。请参阅您的操作系统文档和支持资料，以确定提供了哪些进一步的信息。

应用程序和第三方供应商:

每个应用程序都应该有它自己的记录和诊断文件。这些文件将补充 DB2 信息集，以便使您更准确地了解潜在问题领域。

硬件:

硬件设备通常将信息记录到操作系统错误日志中。但是，有时需要其他信息。在这些情况下，需要确定哪些硬件诊断文件和实用程序可用于环境中的某个硬件。例如，在

DB2 报告页面不正确或某种类型的毁坏时。通常由于磁盘问题而报告此类错误，在这种情况下，需要调查硬件诊断文件。请参阅硬件文档和支持材料，以确定提供了哪些进一步的信息。

总之，要完全了解并评估问题，需要从 DB2、应用程序、操作系统和底层硬件收集所有可用的信息。**db2support** 工具将自动收集您需要的大多数 DB2 和操作系统信息，但您仍然应该了解除此之外可能有助于调查的任何信息。

相关概念:

- 第 3 页的『首次故障数据捕获信息』

系统核心文件（UNIX）

如果程序异常终止，则系统会创建一个核心文件以存储已终止进程的内存映像。诸如内存地址违规、非法指令、总线错误和用户生成的退出信号之类的错误会使核心文件转储。

核心文件名为“core”，位于正在运行应用程序的目录中。注意，系统核心文件不同于 DB2 陷阱文件。

相关概念:

- 第 3 页的『首次故障数据捕获信息』

访问系统核心文件信息（UNIX）

dbx 系统命令帮助您确定哪个函数导致创建系统核心文件。这是一个简单的检查，它将帮助您确定是数据库管理器有错误，或者帮助确定是操作系统还是应用程序错误导致该问题。

先决条件:

必须已安装 **dbx** 命令。

过程:

要确定哪个函数导致发生核心文件转储:

1. 从 UNIX 命令提示符处输入以下命令:

   ```
   dbx program_name core_filename
   ```

   其中 *program_name* 是异常终止的程序的名称，*core_filename* 是包含核心文件转储的文件的名称。

   core_filename 参数是可选的。如果不指定它，则会使用缺省名称“core”。
2. 要获取符号信息，使用“-g”选项编译应用程序。
3. 要结束 **dbx** 命令，在 dbx 提示符下输入 **quit**。
4. 对于 HP-UX 操作系统，对类似函数使用 **xdb** 命令。
5. 打开AIX，使用 **chdev** 命令或 smitty 以确保启用了完整的核心选项。
6. **dbx** 命令提供的功能未在本节中一一描述。有关更多信息，从 UNIX 命令提示符下输入 `man dbx`。

dbx 命令的示例:

以下示例显示如何使用 **dbx** 命令来读取称为 “main” 的程序的核心文件。

1. 在命令提示符下，输入:

   ```
   dbx main
   ```

2. 屏幕上将出现类似如下的输出:

   ```
   dbx version 3.1 for AIX.
   Type 'help' for help.
   reading symbolic information ...
   [using memory image in core]
   segmentation.violation in freeSegments at line 136
   136         (void) shmdt((void *) pcAdress[i]);
   ```

3. 使核心转储的函数的名称为 “freeSegments”。如果函数名以 “db2”、 “sql” 或 “ddcs” 开头，则它可能指示数据库管理器或 DB2 Connect 产品中存在错误。在 dbx 提示符下输入 **where** 以显示故障点的程序路径。

   ```
   (dbx) where
   freeSegments(numSegs = 2, iSetId = 0x2ff7f730, pcAddress = 0x2ff7f758, line
   136
   in "main.c"
   main (0x1, 2ff7f7d4), line 96 in "main.c"
   ```

   在本示例中，freeSegments 的第 136 行出错，它是从 main.c 中的第 96 行调用的。

4. 要结束 **dbx** 命令，在 dbx 提示符下输入 **quit**。

相关概念:

- 第 67 页的『诊断工具（Linux 和 UNIX）』
- 第 15 页的『系统核心文件（UNIX）』

访问事件日志（Windows）

Windows 事件日志还会提供有用的信息。虽然系统事件日志在发生 DB2 崩溃或有关系统资源的其他不明错误时好像最有用，但获取全部三种类型的事件日志仍是值得的:

- 系统
- 应用程序
- 安全性

根据您使用的是 Windows XP、Windows 2003 还是 Windows 2000，用于启动 Windows 事件查看器的方法将会不同。

例如，要在 Windows XP 上打开事件查看器，单击**开始** → **控制面板**。选择**性能和维护**，接着单击**管理工具**，再双击**事件查看器**。

相关概念:

- 第 3 页的『首次故障数据捕获信息』

相关任务:

- 第 17 页的『导出事件日志（Windows）』

导出事件日志（Windows）

可从事件查看器导出两种格式的事件日志:

- .evt 格式，它可以装入回事件查看器（如另一机器上的事件查看器），或者
- 文本格式。

事件查看器格式易于使用，这是因为您可以使用 GUI 来切换时间顺序、过滤特定事件以及提前和推后。

文本文件有一个很大的优点：您可以信任时间戳记！以 .evt 格式导出事件日志时，时间戳记将使用全球标准时间，并且会在查看器中转换为机器的当地时间。如果不够仔细，可能会因为时差错过关键事件。文本文件的搜索也很容易，但是，在将另一台机器上的事件日志装入到事件查看器中之后，它就很容易再次导出为文本格式。

相关概念:

- 第 3 页的『首次故障数据捕获信息』

相关任务:

- 第 16 页的『访问事件日志（Windows）』

访问 Dr. Watson 日志文件（Windows）

Dr. Watson 日志 drwtsn32.log 是系统上发生的所有异常的编年表。尽管 Dr. Watson 在评估整体系统稳定性及描述 DB2 陷阱的历史记录时很有帮助，但 DB2 陷阱文件比 Dr. Watson 日志更有用。缺省路径为 <install_drive>:\Documents and Settings \All Users\Documents\DrWatson

相关概念:

- 第 3 页的『首次故障数据捕获信息』

相关参考:

- 第 66 页的『诊断工具（Windows）』

组合 DB2 数据库和操作系统诊断

在诊断与内存、交换文件、CPU、磁盘存储器和其他资源有关的一些问题时，需要完整地了解给定操作系统管理这些资源的方式。至少在确定与资源有关的问题时需要知道对于每个用户而言，该资源存在的限制及限制程度。（相关限制通常用于 DB2 实例所有者的用户标识。）

以下是需要获取的一些重要配置信息:

- 操作系统补丁级别、已安装软件和升级历史
- CPU 数目
- RAM 量
- 交换和文件高速缓存设置
- 用户数据和文件资源限制及每个用户的进程极限
- IPC 资源限制（消息队列、共享内存段和信号量）
- 磁盘存储器类型

- 机器还有什么功能？DB2 是否争用资源？
- 认证在何处进行？

大多数平台有直接的命令可用来检索资源信息。但是，您很少需要手工获取该信息，原因是 **db2support** 实用程序会收集此数据及更多其他信息。当指定了 **-s** 和 **-m** 选项时，**db2support** 生成的 detailed_system_info.html 文件包含用来收集此信息的许多操作系统命令的语法。

下列练习用于帮助您发现 DB2 诊断文件中的系统配置和用户环境信息。第一个练习让您熟悉运行 **db2support** 实用程序涉及的步骤。后续练习涉及陷阱文件，这些文件会提供更多 DB2 生成的数据，而这些数据在了解用户环境和资源限制时非常有用。

练习 1：运行 **db2support** 命令

1. 使用 **db2start** 命令启动 DB2 实例。
2. 假定您已具有可用的 SAMPLE 数据库，请创建一个目录以存储 **db2support** 的输出。
3. 切换至该目录并发出:

   ```
   db2support <directory> -d sample -s -m
   ```
4. 复查控制台输出，特别是所收集信息的类型。

   在 Windows 上运行时，应该看到类似以下的输出:

   ```
   ...
   正在收集 “系统文件”
         “db2cache.prf”
         “db2cos9402136.0”
         “db2cos9402840.0”
         “db2dbamr.prf”
         “db2diag.bak”
         “db2eventlog.000”
         “db2misc.prf”
         “db2nodes.cfg”
         “db2profile.bat”
         “db2systm”
         “db2tools.prf”
         “HealthRulesV82.reg”
         “db2dasdiag.log”
        ...
   正在收集 “详细的操作系统和硬件信息”
   正在收集 “系统资源信息（磁盘、CPU和内存）”
   正在收集 “操作系统和级别”
   正在收集 “JDK 级别”
   正在收集 “DB2 发行版信息”
   正在收集 “DB2 安装路径信息”
   正在收集 “注册表信息”
   ...
   正在创建最终输出归档
        "db2support.html"
        "db2_sqllib_directory.txt"
        "detailed_system_info.html"
        "db2supp_system.zip"
        "dbm_detailed.supp_cfg"
        "db2diag.log"
   db2support 现在已完成。
   已生成归档文件 “db2support.zip”
   ```
5. 现在使用 Web 浏览器来查看 detailed_system_info.html 文件。在每个系统上标识下列信息:
   - CPU 数目

- 操作系统级别
- 用户环境
- 使用资源限制（UNIX **ulimit** 命令）

练习 2: 查找 DB2 陷阱文件中的环境信息

1. 确保 DB2 实例已启动，然后发出

   ```
   db2pd -stack all
   ```

   调用堆栈放在数据库管理器配置参数 DIAGPATH 中定义的诊断目录的文件中。
2. 在其中一个陷阱文件中查找下列内容:
   - DB2 代码级别
   - 数据段顶部（这是所需的最大专用地址空间）
   - 当前数据大小（这是最大专用地址空间限制）
   - 当前核心大小（这是最大核心文件限制）
   - 信号处理程序（此信息可能不会出现在所有陷阱文件中）
   - 环境变量（此信息可能不会出现在所有陷阱文件中）
   - 映射输出（显示装入的库）

Windows 中的示例陷阱文件（被截断）:

```
...
<DB2TrapFile version="1.0">
<Trap>
<Header>
DB2 build information: DB2 v9.1.0.190 s060121 SQL09010
timestamp: 2006-02-17-14.03.43.846000
uname: S:Windows
comment:
process id: 940
thread id: 3592
</Header>
<SystemInformation>
Number of Processors: 1
Processor Type: x86 Family 15 Model 2 Stepping 4
OS Version: Microsoft Windows XP, Service Pack 2 (5.1)
Current Build: 2600
</SystemInformation>
<MemoryInformation>
<Usage>
Physical Memory:    1023 total,     568 free.
Virtual Memory :    2047 total,    1882 free.
Paging File    :    2461 total,    2011 free.
Ext. Virtual   :       0 free.
</Usage>
</MemoryInformation
<EnvironmentVariables>
<![CDATA[
[e] DB2PATH=C:\Program Files\IBM\SQLLIB
[g] DB2_EXTSECURITY=YES
[g] DB2SYSTEM=MYSRVR
[g] DB2PATH=C:\Program Files\IBM\SQLLIB
[g] DB2INSTDEF=DB2
[g] DB2ADMINSERVER=DB2DAS00
]]></EnvironmentVariables>
```

使 DB2 和系统事件或错误相关:

通常会忽略系统消息和错误日志。如果在问题定义和调查的初始阶段花时间执行一个简单任务，就可以在解决问题时节省几小时、几天甚至几星期的时间。该任务将会比较不同日志中的各个条目，并且记录看起来与时间和各个条目引用的资源相关的所有内容。

虽然并非总是与问题诊断有关，但许多情况下系统日志中会提供最好的线索。如果可将报告的系统问题与 DB2 错误相关，通常就能确定直接导致 DB2 症状的原因。很明显的示例包括磁盘错误、网络错误和硬件错误。并不那么明显的示例包括在不同机器（如域控制器或 NIS 服务器）上报告的错误，机器不同会影响连接时间或认证。

可以调查系统日志以评估稳定性，特别是在全新的系统上报告问题时尤其如此。在常用应用程序中间歇发生陷阱可能表示存在底层硬件问题。

以下是系统日志提供的一些其他信息。

- 重要事件，如重新引导系统的时间
- 系统上发生 DB2 陷阱（及失败的其他软件中的错误、陷阱或异常）的时间
- 内核应急启动、文件系统空间不足和交换空间不足错误（可能导致系统无法创建或派生新进程）

系统日志可帮助您在 db2diag.log 中排除作为考虑因素的崩溃条目。如果在 DB2 管理通知或 DB2 诊断日志中发现崩溃恢复但先前没有任何错误，则 DB2 崩溃恢复可能是系统关闭造成的。

这一关联信息原则扩展至来自任何源的日志和所有可标识用户症状。例如，它在标识和描述来自另一应用程序的日志的关联条目时很有用，即使您不能完整地解释它们也是如此。

总结此信息就能够很完整地了解服务器和发生问题时进行的各种事件。

相关参考:

- 『db2pd - Monitor and troubleshoot DB2 database command』（*Command Reference*）
- 『db2support - Problem analysis and environment collection tool command』（*Command Reference*）

第 2 章 DB2 故障诊断

DB2 故障诊断

通常，故障诊断要求您先找出并确定问题，然后再寻求解决方法。本节将提供关于 DB2 特定功能的故障诊断信息。

对于已知常见问题，找出的解决方法在本节以核对表的形式提供。如果您无法在该核对表中找到解决方法，则可以自己收集其他诊断数据并进行分析，或者将该数据提交给 IBM 软件支持部门进行分析。

分区数据库环境故障诊断

在分区数据库环境中发出命令

在分区数据库环境中，您要发出将在实例中的计算机或在多个数据库分区服务器（节点）上运行的命令。可以使用 **rah** 命令或 **db2_all** 命令来达到此目的。**rah** 命令允许您发出将在实例中的计算机上运行的命令。若想要命令在实例中的多个数据库分区服务器上运行，则运行 **db2_all** 命令。本章节概述了这些命令。以下信息仅适用于分区数据库环境。

注:

1. 在 Linux 和 UNIX 平台上，您的登录 shell 程序可以是 Korn shell 程序或任何其他的 shell 程序；但是，不同 shell 程序处理包含特殊字符的命令所用的方式不同。
2. 另外，在 Linux 和 UNIX 平台上，rah 使用 DB2RSHCMD 注册表变量指定的远程 shell 程序。可以在以下两个远程 shell 程序之间选择: ssh（用于更高的安全性要求）或 rsh（或者对于 HP-UX 是 remsh）。ssh 远程 shell 程序用来防止在 UNIX 操作系统环境中以明文形式传输密码。如果不设置此注册表变量，则使用 rsh（或者对于 HP-UX 是 remsh）。
3. 在 Windows 上，要运行 **rah** 命令或 **db2_all** 命令，您必须使用 Administrators 组成员的用户帐户来登录。

要确定命令的作用域，请参阅 *Command Reference*，它指示命令是在单个的还是所有的数据库分区服务器上运行。如果命令在一个数据库分区服务器上运行，而您想该命令在所有的数据库分区服务器上面运行，则使用 **db2_all**。例外情况是 **db2trc** 命令，它在一台计算机的所有逻辑节点（数据库分区服务器）上运行。如果想要在所有计算机的所有逻辑节点上运行 **db2trc**，则使用 **rah**。

相关概念:

- 『rah 和 db2_all 命令概述』（*《管理指南: 实施》*）
- 『指定 rah 和 db2_all 命令』（*《管理指南: 实施》*）

相关参考:

- 『rah 和 db2_all 命令描述』（*《管理指南: 实施》*）

© Copyright IBM Corp. 1993, 2006

第 3 章 DB2 Connect 故障诊断

问题确定

DB2 Connect 环境涉及到多个软件、硬件和通信产品。通过排除和提炼可用的数据以得出结论（找出发生错误的位置）是确定问题的最好方法。

在收集相关信息之后，根据您选择的适当的主题，转到可参考的章节。

相关概念:

- 第 25 页的『诊断工具』
- 第 23 页的『收集相关信息』
- 第 24 页的『初始连接不成功』
- 第 24 页的『初始连接之后遇到的问题』
- 第 48 页的『跟踪实用程序』

收集相关信息

问题确定包括缩小问题的范围和调查可能的原因。正确的出发点是收集相关信息，确定您知道哪些信息，哪些数据尚未收集，以及可以排除哪些途径。至少要回答下列问题。

- 最初的连接已经成功了吗？
- 硬件运行正常吗？
- 通信路径正常吗？
- 有任何通信网络更改会使先前的目录条目无效吗？
- 已经启动数据库了吗？
- 客户机与 DB2 Connect 工作站之间、DB2 Connect 工作站与主机或 iSeries™ 数据库服务器之间的通信中断了吗？是所有客户机还是一个客户机？
- 利用消息中所返回的消息内容和标记可以确定哪些问题？
- 此时将使用诊断工具来提供任何帮助吗？
- 执行类似任务的其他机器运行正确吗？
- 若这是一个远程任务，则在本地能成功执行它吗？

相关概念:

- 第 25 页的『诊断工具』
- 第 23 页的『问题确定』

© Copyright IBM Corp. 1993, 2006

初始连接不成功

复查下列问题并确保遵循了安装步骤:

1. *成功地完成了安装过程吗？*
   - 所有必备软件产品都可用吗？
   - 内存和磁盘空间足够用吗？
   - 安装了远程客户机支持吗？
   - 完成了通信软件的安装并且没有任何错误情况吗？
2. *对于 UNIX 操作系统，创建了产品的实例吗？*
   - 作为 root 用户，您创建了将作为实例所有者和 sysadm 组的用户和组了吗？
3. *成功地处理了许可证信息吗？（如果适用的话）*
   - 对于 UNIX 操作系统，您编辑了节点锁定文件并输入了由 IBM 提供的密码吗？
4. *是否正确地配置了主机或 iSeries 数据库服务器和工作站通信？*
   - 必须考虑的三个配置:
     a. 主机或 iSeries 数据库服务器配置向服务器标识应用程序请求器。主机或 iSeries 服务器数据库管理系统将具有系统目录条目，这些条目将根据位置、网络协议和安全性来定义请求器。
     b. DB2 Connect 工作站配置向服务器定义客户机成员并向客户机定义主机或 iSeries 服务器。
     c. 客户机工作站配置必须定义工作站的名称和通信协议。
   - 关于未执行最初连接的问题分析包括: 验证 PU（物理单元）名称都是完整的且正确的，或者，对于 TCP/IP 连接，验证是否已经指定了正确的端口号和主机名。
   - 主机或 iSeries 服务器数据库管理员和“网络”管理员都有可用来诊断问题的实用程序。
5. *您是否具有主机或 iSeries 服务器数据库管理系统在使用主机或 iSeries 服务器数据库时需要的权限级别？*
   - 考虑用户的访问权限、表限定符的规则以及期望的结果。
6. *如果尝试使用命令行处理器（CLP）对主机或 iSeries 数据库服务器发出 SQL 语句，是否没有成功吗？*
   - 您是否遵循了将 CLP 绑定至主机或 iSeries 数据库服务器的过程？

相关概念:

- 第 23 页的『问题确定』
- 第 24 页的『初始连接之后遇到的问题』

初始连接之后遇到的问题

提供下列问题作为起始点，以帮助缩小问题的范围。

1. *有任何特殊的或不常见的操作环境吗？*
   - 这是新应用程序吗？
   - 正在使用新过程吗？

- 最近是否执行了可能会影响系统的更改？例如，自从上次成功地运行了应用程序或方案之后，是否更改了任何软件产品或应用程序？
- 对于应用程序，是使用哪个应用程序编程接口（API）来创建该应用程序的？
- 是否有使用该软件或通信 API 的其他应用程序在用户的系统上运行？
- 最近安装了 PTF 吗？若当用户试图在他们的操作系统上使用一个自从安装以来从未使用过的功能部件时发生问题，则应确定 IBM 的最新 PTF 级别，并在安装该功能部件之后装入该级别。

2. *以前发生过此错误吗？*
   - 是否有关于先前的错误状态的已记录的解决方案？
   - 谁是参与者？他们是否能够提供可能操作的深入见解？
3. *您探索过使用通信软件命令来返回有关网络的信息吗？*
   - TCP/IP 可能会从使用 TCP/IP 命令和守护程序时检索到有用的信息。
4. *在 SQLCA（SQL 通信区）中是否返回了有帮助的信息？*
   - 问题处理过程应该包括检查 SQLCODE 和 SQLSTATE 字段的内容的步骤。
   - SQLSTATE 允许应用程序员对 DB2 系列数据库产品的常见错误类进行测试。在分布式关系数据库网络中，此字段可以提供一个公共基础。
5. *在服务器中执行了 DB2START 吗？*另外，对于访问远程服务器的客户机，要确保正确地设置了 DB2COMM 环境变量。
6. *执行同一任务的其他机器能够成功地与服务器相连吗？*试图与服务器相连的客户机数目可能已经达到了最大值。若另一个客户机与服务器断开连接，则先前不能连接的客户机现在能连接吗？
7. *机器是否有正确的寻址？*验证该机器在网络中是否是唯一的。
8. *当远程连接时，已经为客户机授予了正确的权限吗？*可能成功地与实例进行了连接，但是，可能未在数据库级或表级授予权限。
9. *这是第一台与远程数据库进行连接的机器吗？*在分布式环境中，网络之间的路由器或桥接器可能会阻塞客户机与服务器之间的通信。例如，当使用 TCP/IP 时，应确保可以对远程主机执行 PING。

   注: 当从版本 7 的客户机通过版本 9 的 DB2 Connect 服务器向主机发出 PING 命令时，DB2 Connect 将不支持该命令。

相关概念:

- 第 23 页的『问题确定』
- 第 48 页的『跟踪实用程序』

诊断工具

当您遇到问题时，可以使用下列设施:

- 首次故障服务日志（其中诊断信息是以可读格式进行合并和存储的）存储在管理通知日志中。
- 两个日志都可在指定路径中找到:

  在 UNIX 系统上，此文件位于 `/u/db2/sqllib/db2dump/notifyloglevel.nfy`，其中 `db2` 表示实例名。

在 Windows 系统上，此文件位于 x:\sqllib\db2\db2diag.log ，其中 x: 表示逻辑驱动器，db2 表示实例名。

- 对于 Windows 操作系统，可以使用“事件查看器”来查看管理通知日志。
- 跟踪实用程序。
- 对于 Linux 和 UNIX 操作系统，**ps** 命令将关于活动进程的进程状态信息返回到标准输出中。
- 对于 UNIX 操作系统，核心文件是在发生服务器错误时，在当前目录中创建的。它包含已终止的进程的内存映像，可用来确定哪个功能导致该错误。

相关概念:

- 『DB2 Connect 性能故障诊断』（*《DB2 Connect 用户指南》*）
- 第 48 页的『跟踪实用程序』

第 4 章 用于故障诊断的工具

db2dart 工具概述

可以使用 **db2dart** 命令来验证数据库及其对象的体系结构是否正确。还可以使用它来显示数据库控制文件的内容，以便从其他情况下可能无法访问的表中抽取数据。

要显示所有可能的选项，只需发出不带任何参数的 **db2dart** 命令。如果命令行中未显式指定一些需要参数的选项（如表空间标识），则会提示输入它们。

缺省情况下，**db2dart** 实用程序将创建名为 databaseName.RPT 的报告文件。对于单一分区数据库环境，将在当前目录中创建该文件。对于多分区数据库环境，将在诊断目录的子目录中创建该文件。该子目录称为 DART####，其中 #### 是数据库分区号。

db2dart 实用程序通过直接从磁盘中读取数据库中的数据和元数据来访问它们。因此，决不能对仍具有活动连接的数据库运行该工具。如果存在活动连接，则该工具将不知道缓冲池中的页面或内存中的控制结构（此处是举例说明），可能会报告假的错误结果。同样，如果对需要进行崩溃恢复或尚未完成前滚恢复的数据库运行 **db2dart**，由于磁盘上的数据性质不一致，可能会导致类似的不一致情况。

相关参考:

- 『db2dart - Database analysis and reporting tool command』（*Command Reference*）

使用 db2diag 分析 db2diag.log 文件

供数据库和系统管理员使用的主日志文件为管理通知日志。db2diag.log 文件可供 DB2 支持机构用于进行故障诊断。

db2diag 工具用于对 db2diag.log 中提供的大量信息进行过滤和格式化。

示例 1: 按数据库名称过滤 db2diag.log:

如果实例中有若干数据库，并且您只希望显示与数据库 “SAMPLE” 有关的信息，则可以按如下所示过滤 db2diag.log:

```
db2diag -g db=SAMPLE
```

因此，将仅显示包含 “DB: SAMPLE” 的 db2diag.log 记录，如:

```
2006-02-15-19.31.36.114000-300 E21432H406          级别: 错误
PID: 940                       TID: 660            PROC: db2syscs.exe
实例: DB2                      节点: 000           数据库: SAMPLE
APPHDL: 0-1056                 APPID: *LOCAL.DB2.060216003103
函数: DB2 UDB, 基本系统实用程序, sqleDatabaseQuiesce, 探测: 2
消息: ADM7507W  数据库停顿请求成功完成。
```

示例 2: 按进程标识过滤 db2diag.log:

以下命令可用来显示进程标识（PID）为 2200，并且在分区 0、1、2 或分区 3 上运行的进程生成的所有严重错误消息:

© Copyright IBM Corp. 1993, 2006

```
db2diag -g level=Severe,pid=940 -n 0,1,2,3
```

注意，此命令可能以不同方式写入，包括 **db2diag -l severe -pid 2200 -n 0,1,2,3**。还要注意的是，**-g** 选项指定区分大小写的搜索，所以此处 “Severe” 会起作用，但如果使用了 “severe” 则会失败。满足如下要求时，这些命令将成功检索 db2diag.log 记录:

```
2006-02-13-14.34.36.027000-300 I18366H421         级别: 严重
PID: 940                  TID: 660         PROC: db2syscs.exe
实例: DB2                      节点: 000          数据库: SAMPLE
APPHDL: 0-1433               APPID: *LOCAL.DB2.060213193043
函数: DB2 UDB, 数据管理, sqldPoolCreate, 探测: 273
RETCODE: ZRC=0x8002003C=-2147352516=SQLB_BAD_CONTAINER_PATH
          “错误的容器路径”
```

示例 3: 格式化 db2diag 工具输出。

以下命令过滤 2006 年 1 月 1 日之后发生，并且包含分区 0、1 或分区 2 上记录的所有非严重错误和严重错误的所有记录。它会输出匹配的记录，因此时间戳记、分区号和级别将出现在第一行上，而 pid、tid 和实例名将出现在第二行上，之后是错误消息:

```
db2diag -time 2006-01-01 -node "0,1,2" -level "Severe, Error" |
db2diag -fmt "Time: %{ts}
分区: %node Message Level: %{level} \nPid: %{pid}  Tid: %{tid}
实例: %{instance}\nMessage: @{msg}\n"
```

生成的输出示例如下所示:

```
时间: 2006-02-15-19.31.36.099000 分区: 000 消息级别: 错误
Pid: 940 Tid: 40 实例: DB2
消息: ADM7506W 已经请求了数据库停顿。
```

有关更多信息，请发出下列命令:

- **db2diag -help** 提供对选项的简短描述
- **db2diag -h brief** 提供对所有选项的描述，不带示例
- **db2diag -h notes** 提供用法说明和限制
- **db2diag -h examples** 提供一小组示例以帮助您入门
- **db2diag -h tutorial** 提供所有可用选项的示例
- **db2diag -h all** 提供最完整的选项列表

相关概念:

- 第 3 页的『首次故障数据捕获信息』

相关参考:

- 『db2diag - db2diag.log analysis tool command』（*Command Reference*）

使用 db2greg 显示和改变全局注册表（UNIX）

全局注册表在文件 /var/db2/global.reg（在 HP-UX 上为 /var/opt/db2/global.reg）中，且仅存在于 UNIX 和 Linux 平台上。它包含三个不同的记录类型:

- “服务”: 服务记录包含产品级别的信息，如版本和安装路径等等。
- “实例”: 实例记录包含实例级别的信息，如实例名、实例路径、版本和 “引导时启动” 标志等等。
- “变量”: 变量记录包含变量级别的信息，如变量名、变量值和注释。

可使用 **db2greg** 工具来查看（及使用 root 用户权限操作）全局注册表。此工具位于 sqllib/bin 和 bin 下的 install 目录中（以供作为 root 用户登录时使用）。仅当 DB2 客户支持机构要求时您才应使用此工具。

相关参考:

- 第 91 页的『与 IBM 联系』

标识产品的版本和服务级别

db2level 命令帮助您确定 DB2 实例的版本和服务级别（构件级别和修订包级别）。要确定 DB2 实例是否为最新服务级别，比较 db2level 输出和 DB2 支持机构 Web 站点 http://www.ibm.com/software/data/db2/udb/support.html 上的修订包下载页面中的信息。

在 Windows 系统上运行 **db2level** 命令的典型结果为:

```
DB21085I 实例“DB2”使用“32”位和 DB2 代码发行版“SQL09010”，其级别标识为
“01010107”。
参考标记为“DB2 v9.1.0.189”、“n060119”、“”和修订包“0”。
产品安装在“c:\SQLLIB”中，DB2 副本名称为“db2build”。
```

这四个参考标记的组合唯一地标识 DB2 实例的精确服务级别。在与 IBM 支持机构联系以获取帮助时，此信息非常重要。

对于 JDBC 或 SQLJ 应用程序，如果要使用用于 SQLJ 和 JDBC 的 IBM DB2 驱动程序，可通过运行 db2jcc 实用程序来确定驱动程序的级别:

```
db2jcc -version

IBM DB2 JDBC Driver Architecture 2.3.63
```

相关概念:

- 『V9.1 新增内容: 多个 DB2 版本和修订包的共存增强功能（Linux 和 UNIX）』（*《新增内容》*）
- 『V9.1 新增内容: 现在支持多个 DB2 版本和修订包共存（Windows）』（*《新增内容》*）

相关任务:

- 第 32 页的『列示安装在系统上的 DB2 产品（Linux 和 UNIX）』

相关参考:

- 『db2level - Show DB2 service level command』（*Command Reference*）
- 『db2ls - List installed DB2 products and features command』（*Command Reference*）
- 『db2support - Problem analysis and environment collection tool command』（*Command Reference*）

使用 db2look 模拟数据库

如果能够创建结构类似另一数据库的数据库，有时会非常有利。例如，与在生产系统上测试新应用程序或恢复计划相比，创建结构和数据类似的测试系统然后对其进行测试更有意义。这样生产系统不会受到测试的负面性能影响，或者因为错误的应用程序导致的意外数据毁坏而造成的影响。而且，在调查问题（如无效结果、性能问题等）时，会更容易在与生产系统完全相同的测试系统上调试该问题。

可使用 **db2look** 工具来抽取必需的 DDL 语句，这些语句是在一个数据库中再现另一个数据库中的数据库对象所需的。该工具还可生成将统计信息从一个数据库复制至另一个数据库所需的 SQL 语句，以及复制数据库配置、数据库管理器配置和注册表变量所需的语句。因为新数据库可能未包含与原始数据库完全相同的一组数据，但您仍然想要对两个系统选择相同的访问计划，所以这一点是非常重要的。

db2look 工具在 *DB2 Command Reference*中作了详细描述，但您可以通过执行该工具（不带任何参数）来查看选项列表。可使用 **-h** 选项来显示更多详细用法。

使用 db2look 来模拟数据库中的表:

要在数据库中抽取表的 DDL，使用 **-e** 选项。例如，创建 SAMPLE 数据库的副本 SAMPLE2，这样会在新数据库中创建第一个数据库中的所有对象。

```
C:\>db2 create database sample2
DB20000I 成功完成了 CREATE DATABASE 命令。
C:\>db2look -d sample -e > sample.ddl
-- 用户为:
-- 为表创建 DDL
-- 正在自动绑定程序包 ...
-- 绑定成功
-- 正在自动绑定程序包 ...
-- 绑定成功
```

注: 如果还想对用户定义的空间、数据库分区组和缓冲池生成 DDL，则在上述命令中 **-e** 的后面添加 **-l** 标志。将不会抽取缺省数据库分区组、缓冲池和表空间。这是因为在缺省情况下它们已经存在于每个数据库中。如果希望模拟它们，必须手工改变它们。

在文本编辑器中打开 sample.ddl 文件。因为您想要对新数据库执行此文件中的 DDL，所以必须将 CONNECT TO SAMPLE 语句更改为 CONNECT TO SAMPLE2。如果使用了 **-l** 选项，则可能需要改变与表空间命令相关联的路径，这样它们也会指向适当的路径。请抽空看一下文件内容的余下部分。您应该会看到样本数据库中的所有用户表的 CREATE TABLE、ALTER TABLE 和 CREATE INDEX 语句:

```
...
-------------------------------------------------
-- 表 "DB2"."ORG" 的 DDL 语句
-------------------------------------------------

 CREATE TABLE "DB2"."ORG"  (
    "DEPTNUMB" SMALLINT NOT NULL ,
    "DEPTNAME" VARCHAR(14) ,
    "MANAGER" SMALLINT ,
    "DIVISION" VARCHAR(10) ,
    "LOCATION" VARCHAR(13) )
   IN "USERSPACE1" ;
...
```

更改连接语句后，执行以下语句:

```
C:\>db2 -tvf sample.ddl > sample2.out
```

看一下 sample2.out 输出文件 – 每一项都应该已经成功执行。如果发生了错误，错误消息应指示问题所在。修正这些问题并再次执行这些语句。

就像您在输出中看到的那样，将导出所有用户表的 DDL。这是缺省行为，但对于包括的表，还有更具体的其他选项可用。例如，如果希望仅包括 STAFF 和 ORG 表，则使用 **-t** 选项:

```
C:\>db2look -d sample -e -t staff org > staff_org.ddl
```

如果希望仅包括带有模式 DB2 的表，则使用 **-z** 选项:

```
C:\>db2look -d sample -e -z db2 > db2.ddl
```

模拟表的统计信息:

如果测试数据库用于测试性能或调试性能问题，则对两个数据库生成的访问方案应该完全相同。优化器根据统计信息、配置参数、注册表变量和环境变量来生成访问方案。如果两个系统间的这些设置完全相同，则访问方案很可能是相同的。

如果将完全相同的数据装入到两个数据库中，并且对它们两个执行相同的 RUNSTATS 选项，则统计信息也应该完全相同。但是，如果数据库包含不同数据，或者如果测试数据库中只使用数据的子集，则统计信息可能会有显著差别。在这种情况下，可使用 **db2look** 来从生产数据库收集统计信息并将它们放在测试数据库中。通过对可更新表的 SYSSTAT 集合创建 UPDATE 语句并对所有表创建 RUNSTATS 命令来执行此操作。

用于创建统计信息语句的选项为 **-m**。返回至 SAMPLE/SAMPLE2 示例，从 SAMPLE 收集统计信息并将其添加至 SAMPLE2:

```
C:\>db2look -d sample -m > stats.dml
-- 用户为:
-- 以模拟方式运行 db2look
```

像以前一样，必须编辑输出文件以便将 CONNECT TO SAMPLE 语句更改为 CONNECT TO SAMPLE2。而且再看一下文件内容的余下部分，以了解某些 RUNSTATS 和 UPDATE 语句的情况:

```
...
-- 模拟表 ORG
RUNSTATS ON TABLE "DB2"."ORG" ;

UPDATE SYSSTAT.INDEXES
SET NLEAF=-1,
    NLEVELS=-1,
    FIRSTKEYCARD=-1,
    FIRST2KEYCARD=-1,
    FIRST3KEYCARD=-1,
    FIRST4KEYCARD=-1,
    FULLKEYCARD=-1,
    CLUSTERFACTOR=-1,
    CLUSTERRATIO=-1,
    SEQUENTIAL_PAGES=-1,
    PAGE_FETCH_PAIRS='',
    DENSITY=-1,
    AVERAGE_SEQUENCE_GAP=-1,
    AVERAGE_SEQUENCE_FETCH_GAP=-1,
    AVERAGE_SEQUENCE_PAGES=-1,
    AVERAGE_SEQUENCE_FETCH_PAGES=-1,
    AVERAGE_RANDOM_PAGES=-1,
    AVERAGE_RANDOM_FETCH_PAGES=-1,
    NUMRIDS=-1,
    NUMRIDS_DELETED=-1,
    NUM_EMPTY_LEAFS=-1
WHERE TABNAME = 'ORG' AND TABSCHEMA = 'DB2   ';
...
```

使用抽取 DDL 的 **-e** 选项的同时，可使用 **-t** 和 **-z** 选项来指定一组表。

抽取配置参数和环境变量:

优化器根据统计信息、配置参数、注册表变量和环境变量来选择计划。可将统计信息与 **db2look** 一起使用来生成必需的配置更新和设置语句。此操作使用 **-f** 选项完成。例如:

```
c:\>db2look -d sample -f>config.txt
-- 用户为: DB2INST1
-- 正在自动绑定程序包 ...
-- 绑定成功
-- 正在自动绑定程序包 ...
-- 绑定成功
```

config.txt 包含类似如下的输出:

```
-- 此 CLP 文件是使用 DB2LOOK 版本 9.1 创建的
-- 时间戳记: 2/16/2006 7:15:17 PM
-- 数据库名称: SAMPLE
-- 数据库管理器版本: DB2/NT 版本 9.1.0
-- 数据库代码页: 1252
-- 数据库整理顺序为: UNIQUE


CONNECT TO SAMPLE;

------------------------------------------------------
-- 数据库和数据库管理器配置参数
------------------------------------------------------

UPDATE DBM CFG USING cpuspeed 2.991513e-007;
UPDATE DBM CFG USING intra_parallel NO;
UPDATE DBM CFG USING comm_bandwidth 100.000000;
UPDATE DBM CFG USING federated NO;

...

---------------------------------
-- 环境变量设置
---------------------------------


COMMIT WORK;

CONNECT RESET;
```

注: 将仅包括影响 DB2 编译器的那些参数和变量。如果影响编译器的注册表变量设置为其缺省值，则它不会显示在“环境变量设置”下面。

相关概念:

- 『用于对生产数据库建模的统计信息』（《*性能指南*》）

相关参考:

- 『db2look - DB2 statistics and DDL extraction tool command』（*Command Reference*）

列示安装在系统上的 DB2 产品（Linux 和 UNIX）

由于能够在系统上安装 DB2 产品的多个副本，并且能够灵活地在您选择的路径中安装 DB2 产品和功能部件，因此，需要使用一个工具来帮助跟踪已经安装了哪些 DB2 产品及其安装位置。在受支持的 Linux 和 UNIX 操作系统上，**db2ls** 命令将列示安装在系统上的 DB2 产品和功能部件，其中包括 DB2 版本 9 的 HTML 文档。

可以使用 **db2ls** 命令来列示:

- DB2 产品在系统上的安装位置，并且会列示 DB2 产品级别
- 特定安装路径中的所有或特定 DB2 产品和功能部件

先决条件:

至少必须安装了一个 DB2 版本 9 产品，/usr/local/bin 目录中才会存在与 **db2ls** 命令的符号链接。

限制:

db2ls 命令是可用来查询 DB2 产品的唯一方法。*不能*使用 Linux 或 UNIX 操作系统本机实用程序查询 DB2 产品。对于任何现有脚本，只要包含的本机安装实用程序用来查询并作用于 DB2 安装，就需要进行更改。

在 Windows 操作系统上*不能*使用 **db2ls** 命令。

过程:

要列示 DB2 产品在系统上的安装路径并列示 DB2 产品级别，请输入以下命令:

```
db2ls
```

该命令将列示安装在系统上的每个 DB2 产品的下列信息:

- 安装路径
- 级别
- 修订包
- 上次修改 DB2 产品时的安装日期列表。

要列示有关安装在特定路径中的 DB2 产品或功能部件的信息，必须指定 *q* 参数:

```
db2ls -q -b baseInstallDirectory
```

其中

- *q* 参数指定您要查询一个产品或功能部件。此参数是必需的。如果查询了 DB2 版本 8 产品，则会返回空白值。
- *b* 参数指定产品或功能部件的安装目录。如果不是从安装目录运行该命令，则此参数是必需的。

根据所提供的参数，该命令将列示以下信息:

- 安装路径。安装路径只需指定一次，无须对每个功能部件都指定它。
- 将显示以下信息:
  - 已安装功能部件的响应文件标识，或者已安装产品的响应文件标识（如果指定了 *p* 选项）。例如，ENTERPRISE_SERVER_EDITION。
  - 功能部件名称，或者产品名（如果指定了 *p* 选项）。
  - 产品版本、发行版、修改级别和修订包级别（VRMF）。例如，9.1.0.0
  - 修订包（如果有的话）。例如，如果安装了修订包 1，则显示的值是 1。这包括临时修订包，例如修订包 1a。
- 如果任何产品的 VRMF 信息不匹配，则将在输出列表末尾都显示一条警告消息。该消息建议应用此修订包。

相关概念:

- 『同一台计算机上的多个 DB2 副本（Linux 和 UNIX）』（《*安装与配置补充手册*》）

相关任务:

- 『使用 db2_install 或 doce_install 命令安装 DB2 产品（Linux 和 UNIX）』（《*安装与配置补充手册*》）
- 『使用 db2_deinstall 或 doce_deinstall 命令除去 DB2 产品（Linux 和 UNIX）』（《*DB2 服务器快速入门*》）
- 『使用缺省的 DB2 选择向导（Windows）』（《*DB2 服务器快速入门*》）

相关参考:

- 『db2ls - List installed DB2 products and features command』（*Command Reference*）
- 『多个 DB2 副本路线图』（《*管理指南: 实施*》）

使用 db2pd 进行监视和故障诊断

db2pd 工具是一个问题确定工具，它包含 DB2 内存集中的快速即时信息。

该工具不需要获得任何锁存器或使用任何引擎资源就可以收集此信息。因此，在 **db2pd** 收集信息时，有可能（并且预计）会检索到正在更改的信息；这样，数据可能不是十分准确。如果遇到正在更改的内存指针，可使用信号处理程序来防止 **db2pd** 异常终止。这可能会导致输出中出现诸如以下的消息: “正在更改的数据结构已强制终止命令”。虽然如此，该工具对于问题确定却非常有用。在不锁存的情况下收集信息有两个好处: 检索速度更快并且不会争用引擎资源。

如果要在出现特定 SQLCODE、ZRC 代码或 ECF 代码时捕获关于数据库管理系统的信息，则可以使用 **db2pdcfg -catch** 命令完成此操作。捕获到错误时，将启动 db2cos（调出脚本）。db2cos 文件可以自动改变，以便运行解决问题所需的任何 **db2pd** 命令、操作系统命令或任何其他命令。在 UNIX 和 Linux 上，模板文件 db2cos 位于 sqllib/bin 中。在 Windows 操作系统上，db2cos 位于 $DB2PATH\bin 目录中。

以下是使用 **db2pd** 快速确定问题的一组示例。

情况 1: 使用 **db2pdcfg -catch** 选项捕获锁定超时

设定错误捕获设置。可以使用 -911、68、-911 或锁定超时。事实上，如果您事先知道锁定类型或锁定名称，则可以使用 **locktype** 或 **lockname** 子选项滤出不想要的锁定捕获。

```
db2pdcfg -catch locktimeout count=1
错误捕获 #1
SQL 代码: 0
原因码: 0
ZRC: -2146435004
ECF: 0
组件标识: 0
锁定名称: 未设置
锁定类型: 未设置
当前计数: 0
最大计数: 1
位图: 0x4A1
操作: 已启用错误代码捕获标记
操作: 执行 sqllib/db2cos 调出脚本
操作: 在 db2diag.log 中产生堆栈跟踪
```

例如，重现锁定超时错误:

```
SQL0911N 因为死锁或超时，所以当前事务已回滚。
原因码为 "68"。SQLSTATE=40001
```

db2diag.log 中将出现一条相应的消息:

```
2006-02-17-01.28.41.803000-300 I22649H285         级别: 事件
PID: 940                  TID: 3136        PROC: db2syscs.exe
实例: DB2                 节点: 000
函数: DB2 UDB, RAS/PD 组件, pdErrorCatch, 探测: 30
开始: ZRC -2146435004 的错误捕获集

...

2006-02-17-01.40.04.091000-300 I23330H389         级别: 事件
PI: 940                        TID: 2136          PROC: db2syscs.exe
实例: DB2                      节点: 000          数据库: SAMPLE
APPHDL: 0-251                  APPID: *LOCAL.DB2.060217064002
函数: DB2 UDB, 锁管理器, sqlplnfd, 探测: 999
DATA #1: 已预先格式化
捕获到 RC -2146435004。正在转储堆栈跟踪。

2006-02-17-01.40.04.106000-300 I23721H416         级别: 事件
PI: 940                        TID: 2136          PROC: db2syscs.exe
实例: DB2                      节点: 000          数据库: SAMPLE
APPHDL: 0-251                  APPID: *LOCAL.DB2.060217064002
函数: DB2 UDB, 跟踪服务, pdInvokeCalloutScript, 探测: 10
开始: 正在从锁管理器 sqlplnfd 调用 C:\PROGRA~1\IBM\SQLLIB\bin\db2cos.bat

...

2006-02-17-01.40.07.715000-300 I25489H399         级别: 事件
PI: 940                        TID: 2136          PROC: db2syscs.exe
实例: DB2                      节点: 000          数据库: SAMPLE
APPHDL: 0-251                  APPID: *LOCAL.DB2.060217064002
函数: DB2 UDB, 跟踪服务, pdInvokeCalloutScript, 探测: 20
停止: 已完成调用 C:\PROGRA~1\IBM\SQLLIB\bin\db2cos.bat
```

查找 db2cos 输出文件。该文件的位置由数据库管理器配置参数 DIAGPATH 控制。输出文件的内容将随您在 db2cos 文件中输入的命令不同而不同。当 db2cos 文件包含 **db2pd -db sample -locks** 命令时，提供的输出示例如下所示:

```
捕获到锁定超时
Thu Feb 17 01:40:04 EST 2006
实例 DB2 数据库: SAMPLE
分区号: 0
PID: 940
TID: 2136
函数: sqlplnfd
组件: 锁管理器
探测: 999
时间戳记: 2006-02-17-01.40.04.106000
AppID: *LOCAL.DB2...
AppHdl:
<snip>
数据库分区 0 -- 数据库 SAMPLE -- 活动 -- 正常运行 0 天 00:06:53
锁定:
Address TranHdl Lockname Type Mode Sts Owner Dur HldCnt Att Rlse
0x402C6B30 3 000200030000000040000000052 Row ..X W* 3 1 0 0 0x40
```

仅查找 "W*"，因为这是经历超时的锁定。可以使用 **db2pd** 命令提供的所有输出将它映射至事务、应用程序、代理程序或甚至映射至 SQL 语句。可以缩小输出范围或使用

其他命令来收集需要的信息。例如，可以更改 **db2pd** 命令选项以使用 **-locks wait** 选项，后一个选项仅打印具有等待状态的锁定。如果需要 **-app** 和 **-agent** 选项，也可以输入它们。

情况 2：将应用程序映射至动态 SQL 语句

db2pd -applications 命令报告动态 SQL 语句的当前和最后一个锚点标识和语句唯一标识。这允许直接从应用程序映射至动态 SQL 语句。

```
db2pd -app -dyn

应用程序:
Address            AppHandl [nod-index] NumAgents  CoorPid  Status
0x00000002006D2120 780      [000-00780] 1          10615    UOW-Executing

C-AnchID C-StmtUID  L-AnchID L-StmtUID  Appid
163      1          110      1          *LOCAL.jmcmahon.050202200412

动态 SQL 语句:
Address            AnchID StmtUID NumEnv NumVar NumRef NumExe Text
0x0000000220A02760 163    1       2      2      2      1      CREATE VIEW MYVIEW
0x0000000220A0B460 110    1       2      2      2      1      CREATE VIEW YOURVIEW
```

情况 3：监视内存使用情况。

尝试了解内存使用情况时，**db2pd -memblock** 命令非常有用。例如:

```
DBMS 集中的所有内存块。

地址               池标识       池名称     块存活时间 大小（字节） I LOC   文件
0x0780000000740068 62         resynch      2          112          1 1746  1583816485
0x0780000000725688 62         resynch      1          108864       1 127   1599127346
0x07800000001F4348 57         ostrack      6          5160048      1 3047  698130716
0x07800000001B5608 57         ostrack      5          240048       1 3034  698130716
0x07800000001A0068 57         ostrack      1          80           1 2970  698130716
0x07800000001A00E8 57         ostrack      2          240          1 2983  698130716
0x07800000001A0208 57         ostrack      3          80           1 2999  698130716
0x07800000001A0288 57         ostrack      4          80           1 3009  698130716
0x0780000000700068 70         apmh         1          360          1 1024  3878879032
0x07800000007001E8 70         apmh         2          48           1 914   1937674139
0x0780000000700248 70         apmh         3          32           1 1000  1937674139
...
```

接下来是已排序的“性能池”输出:

```
按 ostrack 池的大小排序的内存块:
池标识     池名称    总大小（字节）      总计数         LOC   文件
57         ostrack    5160048             1          3047  698130716
57         ostrack    240048              1          3034  698130716
57         ostrack    240                 1          2983  698130716
57         ostrack    80                  1          2999  698130716
57         ostrack    80                  1          2970  698130716
57         ostrack    80                  1          3009  698130716
ostrack 池的总大小:  5400576 字节

按 apmh 池的大小排序的内存块:
池标识     池名称    总大小（字节）      总计数         LOC   文件
70         apmh       40200               2          121   2986298236
70         apmh       10016               1          308   1586829889
70         apmh       6096                2          4014  1312473490
70         apmh       2516                1          294   1586829889
70         apmh       496                 1          2192  1953793439
70         apmh       360                 1          1024  3878879032
70         apmh       176                 1          1608  1953793439
70         apmh       152                 1          2623  1583816485
```

```
70          apmh        48                  1           914   1937674139
70          apmh        32                  1           1000  1937674139
apmh 池的总大小:  60092 字节
...
```

最后一部分输出对整个 DBMS 集的内存使用者进行排序:

```
DBMS 内存集中的所有内存使用者:
池标识      池名称      总大小（字节）          字节 % 总计数      计数 %  LOC    文件
57          ostrack     5160048              71.90  1           0.07   3047   698130716
50          sqlch       778496               10.85  1           0.07   202    2576467555
50          sqlch       271784               3.79   1           0.07   260    2576467555
57          ostrack     240048               3.34   1           0.07   3034   698130716
50          sqlch       144464               2.01   1           0.07   217    2576467555
62          resynch     108864               1.52   1           0.07   127    1599127346
72          eduah       108048               1.51   1           0.07   174    4210081592
69          krcbh       73640                1.03   5           0.36   547    4210081592
50          sqlch       43752                0.61   1           0.07   274    2576467555
70          apmh        40200                0.56   2           0.14   121    2986298236
69          krcbh       32992                0.46   1           0.07   838    698130716
50          sqlch       31000                0.43   31          2.20   633    3966224537
50          sqlch       25456                0.35   31          2.20   930    3966224537
52          kerh        15376                0.21   1           0.07   157    1193352763
50          sqlch       14697                0.20   1           0.07   345    2576467555
...
```

在 UNIX 和 Linux 上，还可以报告专用内存的内存块。例如:

```
db2pd -memb pid=159770
```

```
专用内存集中的所有内存块。
地址                池标识       池名称      块存活时间 大小（字节）  I LOC    文件
0x0000000110469068 88          private      1          2488          1 172    4283993058
0x0000000110469A48 88          private      2          1608          1 172    4283993058
0x000000011046A0A8 88          private      3          4928          1 172    4283993058
0x000000011046B408 88          private      4          7336          1 172    4283993058
0x000000011046D0C8 88          private      5          32            1 172    4283993058
0x000000011046D108 88          private      6          6728          1 172    4283993058
0x000000011046EB68 88          private      7          168           1 172    4283993058
0x000000011046EC28 88          private      8          24            1 172    4283993058
0x000000011046EC68 88          private      9          408           1 172    4283993058
0x000000011046EE28 88          private      10         1072          1 172    4283993058
0x000000011046F288 88          private      11         3464          1 172    4283993058
0x0000000110470028 88          private      12         80            1 172    4283993058
0x00000001104700A8 88          private      13         480           1 1534   862348285
0x00000001104702A8 88          private      14         480           1 1939   862348285
0x0000000110499FA8 88          private      80         65551         1 1779   4231792244
内存集总大小:  94847 字节

按大小排序的内存块:
池标识      池名称    总大小（字节）      总计数          LOC    文件
88          private    65551               1           1779   4231792244
88          private    28336               12          172    4283993058
88          private    480                 1           1939   862348285
88          private    480                 1           1534   862348285
内存集总大小:  94847 字节
```

情况 4: 确定正在用完表空间的应用程序。

通过使用 **db2pd -tcbstats**，可以标识对一个表执行插入操作的次数。以下是称为 TEMP1 的显式临时表的样本信息:

```
TCB 表信息:
Address            TbspaceID TableID PartID MasterTbs MasterTab TableName  SchemaNm ObjClass
DataSize  LfSize LobSize XMLSize
0x0780000020B62AB0 3         2       n/a    3         2         TEMP1      SESSION  Temp
```

```
966       0      0       0

TCB 表状态:
Address            TableName   Scans   UDI   PgReorgs   NoChgUpdts   Reads   FscrUpdates
Inserts   Updates   Deletes   OvFlReads OvFlCrtes
0x0780000020B62AB0 TEMP1       0       0     0          0            0       0
43968     0         0         0        0
```

然后，可以通过 **db2pd -tablespaces** 命令获取表空间 3 的信息。样本输出如下所示:

```
表空间 3 配置:
Address            Type Content PageSz ExtentSz Auto Prefetch BufID BufIDDisk FSC NumCntrs
MaxStripe  LastConsecPg Name
0x0780000020B1B5A0 DMS  UsrTmp  4096   32       Yes  32       1     1         On  1
0          31           TEMPSPACE2

表空间 3 统计信息:
Address            TotalPgs   UsablePgs  UsedPgs    PndFreePgs FreePgs    HWM
State      MinRecTime NQuiescers
0x0780000020B1B5A0 5000       4960       1088       0          3872       1088
0x00000000 0          0

表空间 3 自动调整大小统计信息:
Address            AS  AR  InitSize    IncSize     IIP MaxSize     LastResize      LRF
0x0780000020B1B5A0 No  No  0           0           No  0           None            No

容器:
地址            容器编号      类型    总页数    可用页数    分割集
容器
0x0780000020B1DCC0 0          File    5000       4960       0
/home/db2inst1/tempspace2a
```

您将注意到通过引用可用页数列并将它与总页数值进行比较所占用的空间。

一旦了解这一点，就可以标识使用表 TEMP1 的动态 SQL 语句:

```
db2pd -db sample -dyn

数据库分区 0 -- 数据库 SAMPLE -- 活动 -- 正常运行 0 天 00:13:06

动态高速缓存:
当前使用的内容                 1022197
堆总大小                       1271398
高速缓存溢出标志               0
引用数                         237
语句插入数目                   32
语句删除数目                   13
变动插入数目                   21
语句数目                       19

动态 SQL 语句:
Address            AnchID StmtUID  NumEnv  NumVar  NumRef  NumExe  Text
0x0000000220A08C40 78     1        2       2       3       2       声明
    未记录全局临时标 temp1 (c1 char(6))
0x0000000220A8D960 253    1        1       1       24      24      插入
    到 session.temp1 值（“TEST”）中
```

最后，可以将它映射至 **db2pd -app** 输出以标识应用程序。

```
应用程序:
Address            AppHandl [nod-index] NumAgents  CoorPid Status
0x0000000200661840 501      [000-00501] 1          11246   UOW-Waiting

C-AnchID C-StmtUID  L-AnchID L-StmtUID  Appid
0        0          253      1          *LOCAL.db2inst1.050202160426
```

db2pd -agent 产生的输出将显示验证时写入的行数。

```
Address            AppHandl [nod-index] AgentPid  Priority  Type DBName
0x0000000200698080 501      [000-00501] 11246     0         Coord SAMPLE

State       ClientPid  Userid   ClientNm Rowsread   Rowswrtn   LkTmOt
Inst-Active 26377      db2inst1 db2bp    22         9588       NotSet
```

如果您怀疑临时表正在占用表空间，则这些步骤会稍有不同。仍然可以使用 **db2pd -tcbstats** 来标识具有最大插入数目的表。以下是隐式临时表的样本信息:

```
TCB 表信息:
Address            TbspaceID TableID PartID MasterTbs MasterTab TableName          SchemaNm
ObjClass         DataSize   ...
0x0780000020CC0D30 1         2       n/a    1         2         TEMP (00001,00002) <30>
<JMC Temp     2470 ...
0x0780000020CC14B0 1         3       n/a    1         3         TEMP (00001,00003) <31>
<JMC Temp     2367 ...
0x0780000020CC21B0 1         4       n/a    1         4         TEMP (00001,00004) <30>
<JMC Temp     1872 ...
TCB 表状态:
Address            TableName          Scans   UDI   PgReorgs   NoChgUpdts Reads   FscrUpdates
Inserts ...
0x0780000020CC0D30 TEMP (00001,00002) 0       0     0          0          0       0
43219 ...
0x0780000020CC14B0 TEMP (00001,00003) 0       0     0          0          0       0
42485 ...
0x0780000020CC21B0 TEMP (00001,00004) 0       0     0          0          0       0
0     ...
```

在此示例中，使用命名约定 “TEMP (TbspaceID, TableID)” 的表中有大量插入。这些是隐式临时表。SchemaNm 列中的值具有命名约定 <AppHandl><SchemaNm>，这使得它可以标识正在工作的应用程序。

然后，可以将这些信息映射至 **db2pd -tablespaces** 产生的输出，以查看表空间 1 的已使用空间。在表空间统计信息中记下已使用的页数与可用页数之间的关系。

```
表空间配置:
Address            Id    Type Content PageSz ExtentSz Auto Prefetch BufID BufIDDisk FSC
NumCntrs MaxStripe LastConsecPg Name
0x07800000203FB5A0 1     SMS  SysTmp  4096   32       Yes  320      1     1         On
10       0         31           TEMPSPACE1

表空间统计信息:
Address            Id    TotalPgs   UsablePgs  UsedPgs    PndFreePgs FreePgs    HWM
State      MinRecTime NQuiescers
0x07800000203FB5A0 1     6516       6516       6516       0          0          0
0x00000000 0          0

表空间自动调整大小统计信息:
Address            Id    AS  AR  InitSize    IncSize     IIP MaxSize     LastResize  LRF
0x07800000203FB5A0 1     No  No  0           0           No  0           None        No

容器:
...
```

然后，可以使用命令 **db2pd -app** 标识应用程序句柄 30 和 31（因为它们出现在 **-tcbstats** 输出中）:

```
应用程序:
Address            AppHandl [nod-index] NumAgents  CoorPid    Status          C-AnchID
C-StmtUID  L-AnchID   L-StmtUID  Appid
0x07800000006FB880 31       [000-00031] 1          4784182    UOW-Waiting     0
```

```
0          107        1          *LOCAL.db2inst1.051215214142
0x07800000006F9CE0 30       [000-00030] 1          8966270    UOW-Executing   107
1          107        1          *LOCAL.db2inst1.051215214013
```

最后，使用 **db2pd -dyn** 命令将它映射至动态 SQL:

```
动态 SQL 语句:
Address            AnchID StmtUID    NumEnv     NumVar     NumRef     NumExe
Text
0x0780000020B296C0 107    1          1          1          43         43
select c1, c2 from test group by c1,c2
```

情况 5: 监视恢复。

命令 **db2pd -recovery** 显示了几个可用于验证是否正在进行恢复的计数器。当前日志和当前 LSN 提供了日志位置。已完成的工作计算目前已完成的字节数。

```
恢复:
恢复状态             0x00000401
当前日志             S0000005.LOG
当前 LSN             000002551BEA
作业类型             ROLLFORWARD RECOVERY
作业标识             7
作业开始时间         (1107380474) Wed Feb  2 16:41:14 2005
作业描述             数据库前滚恢复
调用程序类型         用户
总阶段               2
当前阶段             1

进度:
地址                     阶段号 描述     开始时间                 已完成的工作   总工作
0x0000000200667160 1     向前   Wed Feb  2 16:41:14 2005     2268098 字节    未知
0x0000000200667258 2     向后   NotStarted                    0 字节          未知
```

情况 6: 确定事务正在使用的资源大小

命令 **db2pd -transactions** 提供了锁定数、第一个日志序号（LSN）、最后一个 LSN、已使用的日志空间和保留空间。这对于了解事务行为很有用。

```
事务:
Address            AppHandl [nod-index] TranHdl  Locks  State  Tflag
0x000000022026D980 797      [000-00797] 2        108    WRITE  0x00000000
0x000000022026E600 806      [000-00806] 3        157    WRITE  0x00000000
0x000000022026F280 807      [000-00807] 4        90     WRITE  0x00000000

Tflag2     Firstlsn       Lastlsn        LogSpace   SpaceReserved
0x00000000 0x000001072262 0x0000010B2C8C 4518       95450
0x00000000 0x000001057574 0x0000010B3340 6576       139670
0x00000000 0x00000107CF0C 0x0000010B2FDE 3762       79266

TID            AxRegCnt   GXID
0x000000000451 1          0
0x0000000003E0 1          0
0x000000000472 1          0
```

情况 7: 监视日志使用情况。

命令 **db2pd -logs** 对于监视数据库的日志使用情况很有用。通过观察已写入的页面输出，可以确定是否正在使用日志。

```
日志:
当前日志编号                 2
已写入的页面                 846
方法 1 归档状态              成功
方法 1 下一个要归档的日志    2
```

```
方法 1 首次失败                 不适用
方法 2 归档状态                 成功
方法 2 下一个要归档的日志       2
方法 2 首次失败                 不适用

Address            StartLSN       State      Size  Pages  Filename
0x0000000023001BF58 0x000001B58000 0x00000000 1000  1000   S0000002.LOG
0x0000000023001BE98 0x000001F40000 0x00000000 1000  1000   S0000003.LOG
0x0000000230008F58 0x000002328000 0x00000000 1000  1000   S0000004.LOG
```

使用此输出可以确定两个问题:

1. 如果归档存在问题，则“归档状态”将设置为值“失败”，表示最近的日志归档失败。或者，如果正发生的归档失败导致根本无法归档日志，则将设置“首次失败”。
2. 如果日志归档速度非常慢，则“下一个要归档的日志”值将小于“当前日志编号”值。这可能导致填满日志路径，从而在完全填满日志路径时防止数据库中出现任何数据更改。

情况 8: 查看综合系统（sysplex）列表

如果不使用 **db2pd -sysplex** 命令，则报告综合系统（sysplex）列表的唯一方法是通过 DB2 跟踪。

```
综合系统（sysplex）列表:
别名:              HOST
位置名:            HOST1
计数:              1
IP 地址       端口        优先级    连接状态      PRDID
1.2.34.56     400         1           0            0
```

情况 9: 生成堆栈跟踪

可以使用 **db2pd -stack all** 命令为数据库系统生成堆栈跟踪。如果您怀疑某个进程或线程正在循环或正被挂起，则可能要反复使用此命令。

可以通过发出命令 **db2pd -stack <pid>** 来获取特定进程的当前调用堆栈。例如:

```
db2pd -stack 1335470

  正在尝试转储 pid 1335470 的堆栈跟踪。
  有关陷阱文件，请参阅当前 DIAGPATH。
```

如果需要所有 DB2 进程的调用堆栈，则使用命令 **db2pd -stack all**，例如:

```
db2pd -stack all
尝试转储实例的所有堆栈跟踪。有关陷阱文件，请参阅当前 DIAGPATH。
```

如果您正在使用具有多个物理节点的多分区环境，则通过使用命令 **db2_all ″; db2pd -stack all**″ 可以获取所有分区中的信息。但是，如果分区是同一机器上的所有逻辑分区，则使用 **db2pd -alldbp -stacks** 时的速度会更快。

相关参考:

- 『db2pd - Monitor and troubleshoot DB2 database command』（*Command Reference*）

使用 db2support 收集环境信息

对 DB2 问题收集信息时，您需要运行的最重要的 DB2 实用程序是 **db2support**。

db2support 实用程序用于自动收集所有可用的 DB2 诊断信息和系统诊断信息。它还有一个可选的交互式“问与答”会话，该会话会提出有关问题的详情。

使用 db2support 实用程序可以避免可能的用户错误，这是因为您不必手工输入“GET DATABASE CONFIGURATION FOR <database name>”或“LIST TABLESPACES SHOW DETAIL”之类的命令。而且，您不需要有关要运行的命令或要收集的文件的指示信息，因此收集数据的速度会比较快。

注: db2support 实用程序应该由具有 SYSADM 权限的用户（如实例所有者）运行，以便该实用程序可以收集所有必需的信息而不发生错误。如果没有 SYSADM 权限的用户运行 **db2support**，则在实用程序运行“QUERY CLIENT”或“LIST ACTIVE DATABASE”之类的命令时，可能会产生 SQL 错误（SQL1092）。

如果使用 db2support 实用程序来帮助将信息传送至 IBM 支持机构，则在系统遇到问题时运行 **db2support**。这样工具就会及时地收集信息，例如操作系统性能详细信息。如果在出现问题时无法运行实用程序，您仍可以在问题停止之后发出 **db2support** 命令，因为会自动生成某些首次故障数据捕获（FFDC）诊断文件。

执行 **db2support -h** 会生成完整的命令选项列表。对于调试问题所需的大多数信息的收集而言，以下基本调用通常已经足够（注意，如果使用 **-c** 选项，则该实用程序将建立与数据库的连接）：

```
db2support <output path> -d <database name> -c
```

输出的收集非常方便，并且会存储在压缩的 ZIP 归档 db2support.zip 中，以便可以很轻松地在任何系统上传送和解压缩。

db2support 捕获的信息类型取决于调用命令的方式、是否启动了数据库管理器以及能否连接至数据库。

db2support 实用程序在所有条件下收集以下信息:

- db2diag.log
- 所有陷阱文件
- 锁定列表文件
- 转储文件
- 各种与系统有关的文件
- 各种系统命令的输出
- db2cli.ini

根据情况，db2support 实用程序还有可能收集以下信息:

- 活动日志文件
- 缓冲池和表空间（SQLSPCS.1 和 SQLSPCS.2）控制文件（使用 **-d** 选项）
- db2dump 目录的内容
- 核心文件（**-a** 选项用于收集所有核心文件，**-r** 选项用于仅收集最近的核心文件）
- 扩展系统信息（使用 **-s** 选项）

- 数据库配置设置（使用 **-d** 选项）
- 数据库管理器配置设置文件
- 日志文件头文件（使用 **-d** 选项）
- 恢复历史记录文件（使用 **-d** 选项）

HTML 报告 db2support.html 将总是包括下列信息:
- 问题记录（PMR）编号（如果指定了 **-n**）
- 操作系统和级别（如 AIX 5.1）
- DB2 发行版信息
- 是 32 位还是 64 位环境的指示信息
- DB2 安装路径信息
- db2nodes.cfg 的内容
- CPU 和磁盘数目及内存量
- 实例上的数据库列表
- 注册表信息和环境，包括 PATH 和 LIBPATH
- UNIX 的当前文件系统和节点的磁盘可用空间
- Java™ SDK 级别
- 数据库管理器配置
- 数据库恢复历史记录文件列表
- sqllib 目录的 **ls -lR** 输出（或 Windows 上的相应项）
- LIST NODE DIRECTORY 命令的结果
- LIST ADMIN NODE DIRECTORY 命令的结果
- LIST DCS DIRECTORY 命令的结果
- LIST DCS APPLICATIONS EXTENDED 命令的结果
- 所有已安装软件的列表

指定 **-s** 时，以下信息将出现在 db2support.html 文件中:
- 详细的磁盘信息（分区布局、类型、LVM 信息等等）
- 详细的网络信息
- 内核统计信息
- 固件版本
- 其他特定于操作系统的命令

如果已经启动了 DB2，则 db2support.html 文件包含下列附加信息:
- 客户机连接状态
- 数据库和数据库管理器配置（数据库配置需要 **-d** 选项）
- CLI 配置
- 内存池信息（大小和耗用大小）。如果使用 **-d** 选项，则会收集完整数据
- LIST ACTIVE DATABASES 命令的结果
- LIST DCS APPLICATIONS 命令的结果

如果已指定 **-c** 并且已成功连接至数据库，则 db2support.html 文件包含以下信息:

- 用户表的数目
- 数据库数据的大概大小
- 数据库快照
- 应用程序快照
- 缓冲池信息
- LIST APPLICATIONS 命令的结果
- LIST COMMAND OPTIONS 命令的结果
- LIST DATABASE DIRECTORY 命令的结果
- LIST INDOUBT TRANSACTIONS 命令的结果
- LIST DATABASE PARTITION GROUPS 命令的结果
- LIST DBPARTITIONNUMS 命令的结果
- LIST ODBC DATA SOURCES 命令的结果
- LIST PACKAGES/TABLES 命令的结果
- LIST TABLESPACE CONTAINERS 命令的结果
- LIST TABLESPACES 命令的结果
- LIST DRDA® IN DOUBT TRANSACTIONS 命令的结果

相关参考:

- 『db2support - Problem analysis and environment collection tool command』（*Command Reference*）

跟踪

基本跟踪诊断

如果遇到与 DB2 相关的反复出现并且可再现的问题，则执行跟踪有时能够捕获有关该问题的其他信息。在正常情况下，仅当 DB2 客户支持机构要求时，才应该使用跟踪。执行跟踪的过程包括设置跟踪工具、再现错误与收集数据。

跟踪收集的信息量增长得相当快。当您执行跟踪时，请尽可能地只捕获错误状态并避免任何其他活动。在执行跟踪时，请尽可能地使用最省力的方案来再现问题。

执行跟踪的过程通常对 DB2 实例的行为具有全局效果。性能的降低程度取决于问题的类型以及使用了多少资源来收集跟踪信息。

在请求跟踪时，DB2 客户支持机构应提供下列信息:

- 简单的逐步过程
- 进行每次跟踪的位置的说明
- 跟踪内容的说明
- 请求跟踪的理由的说明
- 回退过程（例如，如何禁用所有跟踪）

尽管您应该听从 DB2 客户支持机构有关获取哪些跟踪的建议，但以下仍然提供要求您获取特定跟踪时的一些一般准则:

- 如果安装期间发生问题，并且缺省安装日志不足以确定问题原因，则比较适合进行安装跟踪。
- 如果在其中一个 GUI（图形用户界面）工具中发生问题，并且通过 DB2 命令窗口中的一些显式命令成功地执行了某些操作，则比较适合进行控制中心跟踪。注意，这只会使用可从控制中心启动的工具捕获问题。
- 如果 CLI 应用程序中出现问题，并且该问题不能在该应用程序之外再现，则比较适合进行 CLI 跟踪。
- 如果 JDBC 应用程序中出现问题，并且该问题不能在该应用程序之外再现，则比较适合进行 JDBC 跟踪。
- 如果该问题与要在 DRDA 层进行通信的信息直接有关，则进行 DRDA 跟踪比较适合。
- 对于可进行跟踪的所有其他情况，最适合进行 DB2 跟踪。

在诊断错误时，跟踪信息并不总是对您有帮助。例如，在下列情况下可能无法捕获错误情况:

- 您指定的跟踪缓冲区大小不够大，无法存放完整的跟踪事件集，当跟踪停止写入文件或合并时，丢失了有用的信息。
- 跟踪的方案没有重新创建错误状态。
- 重新创建了错误状态，但对问题发生位置的假定不正确。例如，跟踪是在客户机工作站上收集到的，但实际错误发生在服务器上。

相关概念:

- 第 45 页的『使用 db2trc 获取 DB2 跟踪』
- 第 48 页的『DRDA 跟踪文件』

DB2 跟踪

使用 db2trc 获取 DB2 跟踪

db2trc 命令控制随 DB2 提供的跟踪工具。跟踪工具记录有关操作的信息并将此信息格式化为可读格式。

记住，运行跟踪时会增加开销，所以启用跟踪工具可能会影响系统性能。

通常 DB2 支持机构和开发团队使用 DB2 跟踪确定问题。您可以运行跟踪来获取有关正在调查的问题的信息，但如果您不了解 DB2 源代码的话，它的用途将十分有限。

即使只要求您获取跟踪文件，您也应该知道如何正确地启动跟踪以及如何转储跟踪文件。

注: 您需要 SYSADM、SYSCTRL 或 SYSMAINT 权限的其中一种权限来使用 **db2trc**。要大致了解可用的选项，执行不带任何参数的 **db2trc** 命令:

```
C:\>db2trc
用法: db2trc (chg|clr|dmp|flw|fmt|inf|off|on) options
```

有关特定 **db2trc** 命令参数的更多信息，请使用 **-u** 选项。例如，要查看有关启动跟踪的更多信息，请执行以下命令:

```
db2trc on -u
```

这会提供有关可在启动 DB2 跟踪时指定的所有附加选项（标注为“工具”）的信息。

在启动跟踪时需要知道的最重要选项是 -L。它指定用于存储跟踪信息的内存缓冲区的大小。缓冲区大小可以字节或兆字节为单位来指定。（要指定兆字节，则在值后面追加“M”或“m”。跟踪缓冲区大小必须是 2 兆字节的幂。如果指定的大小不符合此要求，则缓冲区大小将自动舍入为最接近的 2 的幂。

如果缓冲区太小，则信息可能会丢失。在缺省情况下，如果缓冲区变满，则只会保留最新的跟踪信息。如果缓冲太大，可能难以将文件发送至 DB2 支持团队。

如果跟踪时间相对较短的操作（如数据库连接），则大概 8MB 大小通常就已足够:

```
C:\> db2trc on -l 8M
已启动跟踪
```

但是，如果跟踪规模较大的操作或者同时在进行大量工作，则可能需要较大的跟踪缓冲区。

在大多数平台上，跟踪可按如上所述随时打开并工作。但是，有些特定情况需要注意:

1. 在多数据库分区系统上，必须对每个物理（相对于逻辑）数据库分区运行跟踪。
2. 在 HP-UX 上，Linux 和 Solaris 平台上，如果在启动实例后关闭跟踪，则不管指定大小如何都会使用非常小的缓冲区。要在这些平台上有效地运行跟踪，在启动实例前启动跟踪并在以后必要时清除它。

相关概念:

- 第 46 页的『转储 DB2 跟踪文件』
- 第 47 页的『格式化 DB2 跟踪文件』

相关参考:

- 『db2trc - Trace command』（*Command Reference*）

转储 DB2 跟踪文件

使用 on 选项启用跟踪工具后，将跟踪实例所作的所有后续工作。

跟踪运行时，可使用 clr 选项来清除跟踪缓冲区。将除去跟踪缓冲区中的所有现有信息。

```
C:\>db2trc clr
已清除跟踪
```

要跟踪的操作完成后，使用后跟跟踪文件名的 dmp 选项将内存缓冲池转储至磁盘。例如:

```
C:\>db2trc dmp trace.dmp
跟踪已转储至文件
```

跟踪缓冲区转储至磁盘后，跟踪工具将继续运行。要关闭跟踪，使用 off 选项:

```
C:\>db2trc off
已关闭跟踪
```

相关概念:

- 第 47 页的『格式化 DB2 跟踪文件』
- 第 45 页的『使用 db2trc 获取 DB2 跟踪』

相关参考:

- 『db2trc - Trace command』（*Command Reference*）

格式化 DB2 跟踪文件

db2trc dmp 命令创建的转储文件为二进制格式，并且不可读取。

要验证是否可读取跟踪文件，请对二进制跟踪文件进行格式化以显示流量控制并将格式化输出发送至空设备。下面的示例显示了用于执行此任务的命令:

```
db2trc flw example.trc nul
```

其中 example.trc 使使用 **dmp** 选项生成的二进制文件。

此命令的输出将显式地指出读取文件是否有问题，以及跟踪是否已合并。

此时，可将转储文件发送至 DB2 支持机构。他们会根据您的 DB2 服务级别来对其进行格式化。但有时可能会要求您在发送转储文件之前将其转换为 ASCII 格式。这是通过 **flw** 和 **fmt** 选项完成的。必须提供二进制转储文件的名称及想要创建的 ASCII 文件的名称:

```
C:\>db2trc flw trace.dmp trace.flw
C:\Temp>db2trc flw trace.dmp trace.flw
总跟踪记录数: 18854
截断跟踪: NO
合并跟踪: NO
格式化的跟踪记录数: 1513（pid: 2196 tid 2148 节点: -1）
格式化的跟踪记录数: 100（pid: 1568 tid 1304 节点: 0）
...

C:\>db2trc fmt trace.dmp trace.fmt
C:\Temp>db2trc fmt trace.dmp trace.fmt
截断跟踪: NO
合并跟踪: NO
总跟踪记录数: 18854
格式化的跟踪记录数: 18854
```

如果此输出指示“合并跟踪”为“YES”，则表示跟踪缓冲区不够大，无法包含跟踪时间段收集的所有信息。根据情况，合并跟踪也许是可行的。如果您关心的是最新信息（除非指定了 **-i** 选项，否则这是缺省情况下保留的信息），则跟踪文件中的内容可能已经足够。但是，如果您关心的是跟踪时间段开始时发生的情况，或者如果您关心的是发生的所有情况，则您可能想要使用更大的跟踪缓冲区重做该操作。

要注意的另一件事是: 在 Linux 和 UNIX 操作系统上，DB2 在因为严重错误而关闭实例时会自动将跟踪缓冲区转储至磁盘。因此，如果实例异常结束时启用了跟踪，则会在诊断目录中创建一个文件，其名称为 db2trdmp.###，其中 ### 是节点号。Windows 平台上不会发生这种情况。在这些情况下，您必须手工转储跟踪。

为进行总结，以下是 **db2trc** 命令的常用顺序示例:

```
db2trc on -l 8M
   db2trc clr
   <Execute problem recreation commands>
   db2trc dump db2trc.dmp
   db2trc off
   db2trc flw db2trc.dmp <filename>.flw
   db2trc fmt db2trc.dmp <filename>.fmt
   db2trc fmt -c db2trc.dmp <filename>.fmtc
```

相关概念:

- 第 46 页的『转储 DB2 跟踪文件』
- 第 45 页的『使用 db2trc 获取 DB2 跟踪』

相关参考:

- 『db2trc - Trace command』（*Command Reference*）

DRDA 跟踪

DRDA 跟踪文件

在分析 DRDA 跟踪之前，需要了解 DRDA 是数据和通信结构的定义的开放式标准。例如，DRDA 包含有关数据传输的组织方式以及有关该信息的通信方式的一组规则。这些规则是在以下参考手册中定义的:

- DRDA V3 Vol. 1: 分布式关系数据库体系结构™
- DRDA V3 Vol. 2: 格式化数据对象内容体系结构
- DRDA V3 Vol. 3: 分布式数据管理体系结构

这些手册的 PDF 版本可从 www.opengroup.org 获取。

db2drdat 实用程序记录 DRDA Application Requestor（AR）与 DB2 DRDA Application Server（AS）之间（如 DB2 Connect 与主机或 iSeries 数据库服务器之间）的数据交换。

相关概念:

- 『DB2 Connect 和 DRDA』（*《DB2 Connect 用户指南》*）
- 第 48 页的『跟踪实用程序』

相关参考:

- 『db2drdat - DRDA trace command』（*Command Reference*）

跟踪实用程序

db2drdat 实用程序记录 DB2 Connect 服务器（代表数据库客户机）与主机或 iSeries 数据库服务器之间交换的数据。

作为数据库管理员（或应用程序开发者），您可能会发现了解此数据流如何工作是很有用的，因为这些知识可以帮助您确定特定问题的起源。例如，如果您对主机或 iSeries 数据库服务器发出了 `CONNECT TO` 数据库语句，但是该命令失败了并且您接收到一个不成功的返回码。如果您明确知道哪些信息被传送到主机或 iSeries 数据库服务器管理系统，则就可能确定故障的原因，即使返回码信息是一般信息。很多故障是由简单的用户错误造成的。

来自 `db2drdat` 的输出列示了 DB2 Connect 工作站与主机或 iSeries 数据库服务器管理系统之间交换的数据流。发送至主机或 iSeries 数据库服务器的数据被标记为 SEND BUFFER，而从主机或 iSeries 数据库服务器接收的数据被标记为 RECEIVE BUFFER。

如果接收缓冲区中包含 SQLCA 信息，则它后面将紧接此数据的已格式化解释并标记为 SQLCA。SQLCA 的 SQLCODE 字段是主机或 iSeries 数据库服务器所返回的*未映射的*值。在文件中，发送缓冲区和接收缓冲区是按从最旧到最新的顺序来排列的。每个缓冲区都具有:

- 进程标识。
- SEND BUFFER、RECEIVE BUFFER 或 SQLCA 标号。缓冲区中的第一个 DDM 命令或对象被标记为 DSS TYPE。

在发送缓冲区和接收缓冲区中的其他数据被分成五列，包括:

- 字节数。
- 第 2 列和第 3 列表示在两个系统之间所交换的DRDA 数据流，采用 ASCII 或 EBCDIC 格式。
- 第 2 列和第 3 列的 ASCII 表示。
- 第 2 列和第 3 列的 EBCDIC 表示。

相关概念:

- 第 49 页的『跟踪输出』
- 第 50 页的『跟踪输出文件分析』

相关参考:

- 『db2drdat - DRDA trace command』（*Command Reference*）

跟踪输出

db2drdat 实用程序将下列信息写入到*跟踪文件*中:

- -r
  - DRDA 应答 / 对象的类型
  - 接收缓冲区
- -s
  - DRDA 请求的类型
  - 发送缓冲区
- -c
  - SQLCA
- TCP/IP 错误信息
  - 接收函数返回码
  - 严重性
  - 使用的协议
  - 使用的 API
  - 函数
  - 错误号

注:

1. 出口码的值为零，指示成功地完成了命令，非零值指示没有成功地完成命令。
2. 返回的字段将随使用的 API 不同而不同。
3. 返回的字段将随运行 DB2 Connect 的平台不同而不同，即使对于同一 API 也是如此。
4. 如果 **db2drdat** 命令将输出发送至已存在的文件，则将擦除旧文件，除非文件的许可权不允许擦除它。

相关概念:

- 第 50 页的『跟踪输出文件分析』
- 第 48 页的『跟踪实用程序』

相关参考:

- 『db2drdat - DRDA trace command』（*Command Reference*）

跟踪输出文件分析

在 **db2drdat** 跟踪中捕获到下列信息:

- 客户机应用程序的进程标识（PID）
- 在数据库连接服务（DCS）目录中编目的 RDB_NAME
- DB2 Connect CCSID
- 主机或 iSeries 数据库服务器 CCSID
- DB2 Connect 系统正在与其通信的主机或 iSeries 数据库服务器管理系统

首个缓冲区中包含发送到主机或 iSeries 数据库服务器管理系统的“交换服务器属性”（EXCSAT）和“访问 RDB”（ACCRDB）命令。它将这些命令作为 CONNECT TO 数据库命令的结果来发送。下一个缓冲区中包含 DB2 Connect 从主机或 iSeries 数据库服务器管理系统中接收到的应答。它包含“交换服务器属性应答数据”（EXCSATRD）和“访问 RDB 应答消息”（ACCRDBRM）。

EXCSAT

EXCSAT 命令包含由“服务器名”（SRVNAM）对象指定的客户机的工作站名，按照 DDM 规范，它是代码点 X'116D'。EXCSAT 命令在第一个缓冲区中。在 EXCSAT 命令内，一旦除去了 X'116D'，值 X'9481A292'（按 CCSID 500 编码）就被转换为 *mask*。

EXCSAT 命令还包含 EXTNAM（外部名）对象，通常将该对象放在主机或 iSeries 数据库管理系统上的诊断信息中。它由 20 个字节的应用程序标识、后接 8 个字节的进程标识（或者是 4 个字节的进程标识和 4 个字节的线程标识）组成。它由代码点 X'115E' 表示，在此示例中，其值为 db2bp 并用空格填充，后接 000C50CC。在 Linux 或 UNIX 数据库客户机上，可以将此值与 **ps** 命令相关联，该命令将与活动进程有关的进程状态信息返回到标准输出中。

ACCRDB

ACCRDB 命令包含 RDBNAM 对象中的 RDB_NAME，它是代码点 X'2110'。在第一个缓冲区中，ACCRDB 命令跟在 EXCSAT 命令后面。在 ACCRDB 命令内，一旦除去了 X'2110'，值 X'E2E3D3C5C3F1' 就会被转换为 STLEC1。这对应于 DCS 目录中的目标数据库名称字段。

记帐字符串具有代码点 X'2104'。

通过在 ACCRDB 命令中查找具有代码点 X'119C' 的 CCSID 对象 CCSIDSBC（用于单字节字符的 CCSID），就可显示为 DB2 Connect 工作站配置的代码集。在此示例中，CCSIDSBC 为 X'0333'，代码集为 819。

在 ACCRDB 命令中还存在附加对象 CCSIDDBC（双字节字符的 CCSID）和 CCSIDMBC（混合字节字符的 CCSID），它们分别具有代码点 X'119D' 和 X'119E'。在此示例中，CCSIDDBC 为 X'04B0'，代码集为 1200；CCSIDMBC 为 X'0333'，代码集为 819。

EXCSATRD 和 ACCRDBRM

CCSID 值也是从主机或 iSeries 数据库服务器的第二个缓冲区内的 “访问 RDB 应答消息”（ACCRDBRM）中返回的。此缓冲区中包含 EXCSATRD，后接 ACCRDBRM。样本输出文件中包含主机或 iSeries 数据库服务器系统的两个 CCSID 值。这两个值分别为 1208（对于单字节字符和混合字节字符）和 1200（对于双字节字符）。

如果 DB2 Connect 不能识别从主机或 iSeries 数据库服务器返回的代码页，则将对用户返回 SQLCODE -332 以及源和目标代码页。如果主机或 iSeries 数据库服务器不识别从 DB2 Connect 发送的代码集，则它将返回 VALNSPRM（不支持参数值，DDM 代码点为 X'1252'），它将对用户转换为 SQLCODE -332。

ACCRDBRM 还包含参数 PRDID（特定产品标识，具有代码点 X'112E'）。值为 X'C4E2D5F0F8F0F1F5'，用 EBCDIC 表示为 DSN08015。按照标准，DSN 为 DB2 通用数据库 z/OS 和 OS/390 版。还指示了版本号。ARI 是 DB2 服务器 VSE 和 VM 版，SQL 是 DB2 数据库或 DB2 Connect，QSQ 是 DB2 UDB iSeries 版。

相关概念:

- 第 49 页的『跟踪输出』
- 第 48 页的『跟踪实用程序』

相关参考:

- 『db2drdat - DRDA trace command』（*Command Reference*）
- 第 57 页的『DRDA 跟踪的后续缓冲区信息』
- 第 51 页的『跟踪输出文件样本』

跟踪输出文件样本

下列各图显示了样本输出，说明在 DB2 Connect 工作站与主机或 iSeries 数据库服务器之间交换的一些 DRDA 数据流。从用户的角度来看，已经使用命令行处理器（CLP）发出了 CONNECT TO 数据库命令。

图 1 使用 DB2 Connect 企业版版本 9.1 和 DB2 UDB z/OS 版版本 8 来进行 TCP/IP 连接。

```
1 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 0 probe 100
 bytes 16

 Data1  (PD_TYPE_UINT,8) unsigned integer:
 233
```

图 1. 跟踪输出的示例（TCP/IP 连接）(1/20)

```
2 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 19532 probe 1177
 bytes 250

 SEND BUFFER(AR):

           EXCSAT RQSDSS                     (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  00C3D041000100BD 1041007F115E8482   ...A.....A...^..  .C}........".;db
  0010  F282974040404040 4040404040404040   ...@@@@@@@@@@@@@  2bp
  0020  4040F0F0F0C3F5F0 C3C3F0F0F0000000   @@.............    000C50CC000...
  0030  0000000000000000 0000000000000000   ................  ................
  0040  0000000000000000 000000000060F0F0   ............`..  ............-00
  0050  F0F1A2A495404040 4040404040404040   .....@@@@@@@@@@@  01sun
  0060  4040404040404040 4040404040404040   @@@@@@@@@@@@@@@@
  0070  C4C5C3E5F8404040 F0A2A49540404040   .....@@@....@@@@  DECV8   0sun
  0080  4040404040404040 4000181404140300   @@@@@@@@@.......           .......
  0090  0724070008147400 05240F0008144000   .$....t..$....@.  .............. .
  00A0  08000E1147D8C4C2 F261C1C9E7F6F400   ....G....a......  .....QDB2/AIX64.
  00B0  08116D9481A29200 0C115AE2D8D3F0F9   ..m.......Z.....  .._mask...]SQL09
  00C0  F0F0F0                              ...               000

           ACCSEC RQSDSS                     (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  0026D00100020020 106D000611A20003   .&..... .m......  ..}......_...s..
  0010  00162110E2E3D3C5 C3F1404040404040   ..!.......@@@@@@  ....STLEC1
  0020  404040404040                        @@@@@@
```

图 1. 跟踪输出的示例（TCP/IP 连接） (2/20)

```
3 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110546200 probe 100
 bytes 12

 Data1  (PD_TYPE_UINT,4) unsigned integer:
 105
```

图 1. 跟踪输出的示例（TCP/IP 连接） (3/20)

```
4 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110549755 probe 1178
 bytes 122

 RECEIVE BUFFER(AR):

           EXCSATRD OBJDSS                   (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  0059D04300010053 1443000F115EE5F8   .Y.C...S.C...^..  ..}.........;V8
  0010  F1C14BE2E3D3C5C3 F100181404140300   ..K.............  1A.STLEC1.......
  0020  0724070007147400 05240F0007144000   .$....t..$....@.  .............. .
  0030  0700081147D8C4C2 F20014116DE2E3D3   ....G.......m...  .....QDB2..._STL
  0040  C5C3F14040404040 4040404040000C11   ...@@@@@@@@@@...  EC1          ...
  0050  5AC4E2D5F0F8F0F1 F5                Z........         ]DSN08015

           ACCSECRD OBJDSS                   (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  0010D0030002000A 14AC000611A20003   ................  ..}.........s..
```

图 1. 跟踪输出的示例（TCP/IP 连接） (4/20)

```
5 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110656806 probe 100
 bytes 16

 Data1  (PD_TYPE_UINT,8) unsigned integer:
 233
```

图 1. 跟踪输出的示例（TCP/IP 连接） (5/20)

```
6 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110659711 probe 1177
 bytes 250

 SEND BUFFER(AR):

           SECCHK RQSDSS                    (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  003CD04100010036 106E000611A20003   .<.A...6.n......  ..}......>...s..
  0010  00162110E2E3D3C5 C3F1404040404040   ..!.......@@@@@@  ....STLEC1
  0020  4040404040040000C 11A1D9858799F485   @@@@@@..........        ....Regr4e
  0030  A599000A11A09585 A6A39695           ............      vr....newton

           ACCRDB RQSDSS                    (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  00ADD001000200A7 20010006210F2407   ........ ...!.$.  ..}....x........
  0010  00172135C7F9F1C1 F0C4F3C14BD7C1F8   ..!5........K...  ....G91A0D3A.PA8
  0020  F806030221064600 162110E2E3D3C5C3   ....!.F..!......  8..........STLEC
  0030  F140404040404040 4040404040000C11   .@@@@@@@@@@@@...  1            ...
  0040  2EE2D8D3F0F9F0F0 F0000D002FD8E3C4   ............/...  .SQL09000....QTD
  0050  E2D8D3C1E2C30016 00350006119C0333   .........5.....3  SQLASC..........
  0060  0006119D04B00006 119E0333003C2104   ...........3.
```

图 1. 跟踪输出的示例（TCP/IP 连接） (6/20)

```
7 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259908001 probe 100
 bytes 12

 Data1  (PD_TYPE_UINT,4) unsigned integer:
 176
```

图 1. 跟踪输出的示例（TCP/IP 连接） (7/20)

```
8 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259911584 probe 1178
 bytes 193

 RECEIVE BUFFER(AR):

           SECCHKRM RPYDSS                   (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  0015D0420001000F 1219000611490000   ...B.........I..  ..}.............
  0010  000511A400                          .....             ...u.

           ACCRDBRM RPYDSS                   (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  009BD00200020095 2201000611490000   ........"....I..  ..}....n........
  0010  000D002FD8E3C4E2 D8D3F3F7F0000C11   .../............  ....QTDSQL370...
  0020  2EC4E2D5F0F8F0F1 F500160035000611    ............5...  .DSN08015.......
  0030  9C04B80006119E04 B80006119D04B000    ................  ................
  0040  0C11A0D5C5E6E3D6 D5404000006212524   .........@@..!%$  ...NEWTON  .....
  0050  34001E244E000624 4C00010014244D00   4..$N..$L....$M.  ....+...<.....(.
  0060  06244FFFFF000A11 E8091E768301BE00   .$O........v....  ..!.....Y...c...
  0070  2221030000000005 68B3B8C7F9F1C1F0   "!......h.......  ...........G91A0
  0080  C4F3C1D7C1F8F840 4040400603022106   .......@@@...!.  D3APA88    .....
  0090  46000A11E8091E76 831389            F......v...       ....Y...c.i
```

图 1. 跟踪输出的示例（TCP/IP 连接） (8/20)

```
9 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364420503 probe 100
 bytes 16

 Data1  (PD_TYPE_UINT,8) unsigned integer:
 10
```

图 1. 跟踪输出的示例（TCP/IP 连接） (9/20)

```
10 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364440751 probe 1177
 bytes 27

 SEND BUFFER(AR):

           RDBCMM RQSDSS                     (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  000AD00100010004 200E               ........ .        ..}.......
```

图 1. 跟踪输出的示例（TCP/IP 连接） (10/20)

```
11 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475009631 probe 100
 bytes 12

 Data1  (PD_TYPE_UINT,4) unsigned integer:
 54
```

图 1. 跟踪输出的示例（TCP/IP 连接） (11/20)

```
12 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475014579 probe 1178
 bytes 71

 RECEIVE BUFFER(AR):

           ENDUOWRM RPYDSS                  (ASCII)           (EBCDIC)
          0 1 2 3 4 5 6 7  8 9 A B C D E F  0123456789ABCDEF  0123456789ABCDEF
  0000  002BD05200010025 220C000611490004  .+.R...%"....I..  ..}.............
  0010  00162110E2E3D3C5 C3F1404040404040  ..!.......@@@@@@  ....STLEC1
  0020  4040404040400005 211501            @@@@@@..!..             .....

           SQLCARD OBJDSS                   (ASCII)           (EBCDIC)
          0 1 2 3 4 5 6 7  8 9 A B C D E F  0123456789ABCDEF  0123456789ABCDEF
  0000  000BD00300010005 2408FF            ........$..       ..}........
```

图 1. 跟踪输出的示例（TCP/IP 连接） (12/20)

```
13 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721710319 probe 100
 bytes 16

 Data1  (PD_TYPE_UINT,8) unsigned integer:
 126
```

图 1. 跟踪输出的示例（TCP/IP 连接） (13/20)

```
14 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721727276 probe 1177
 bytes 143

 SEND BUFFER(AR):

           EXCSQLIMM RQSDSS                 (ASCII)           (EBCDIC)
          0 1 2 3 4 5 6 7  8 9 A B C D E F  0123456789ABCDEF  0123456789ABCDEF
  0000  0053D0510001004D 200A00442113E2E3  .S.Q...M ..D!...  ..}....(......ST
  0010  D3C5C3F140404040 4040404040404040  ....@@@@@@@@@@@@  LEC1
  0020  D5E4D3D3C9C44040 4040404040404040  ......@@@@@@@@@@  NULLID
  0030  4040E2D8D3C3F2C6 F0C1404040404040  @@........@@@@@@    SQLC2F0A
  0040  4040404041414141 41484C5600CB0005  @@@@AAAAAAHLV....      ......<.....
  0050  2105F1                             !..               ..1

           SQLSTT OBJDSS                    (ASCII)           (EBCDIC)
          0 1 2 3 4 5 6 7  8 9 A B C D E F  0123456789ABCDEF  0123456789ABCDEF
  0000  002BD00300010025 2414000000001B64  .+.....%$......d  ..}.............
  0010  656C657465206672 6F6D206464637375  elete from ddcsu  .%......?_......
  0020  73312E6D79746162 6C65FF            s1.mytable.       ..._`./.%..
```

图 1. 跟踪输出的示例（TCP/IP 连接） (14/20)

```
15 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832901261 probe 100
 bytes 12

 Data1  (PD_TYPE_UINT,4) unsigned integer:
 102
```

图 1. 跟踪输出的示例（TCP/IP 连接） (15/20)

```
16 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832906528 probe 1178
 bytes 119

 RECEIVE BUFFER(AR):

           SQLCARD OBJDSS                    (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  0066D00300010060 240800FFFFFF3434   .f.....`$.....44  ..}....-........
  0010  3237303444534E58 4F544C2000FFFFFE   2704DSNXOTL ....  ......+.!.<.....
  0020  0C00000000000000 00FFFFFFFF000000   ................  ................
  0030  0000000000572020 2057202020202020   .....W   W        ................
  0040  001053544C454331 2020202020202020   ..STLEC1          ....<...........
  0050  2020000F44444353 5553312E4D595441     ..DDCSUS1.MYTA  ............(...
  0060  424C450000FF                         BLE...            .<....
```

图 1. 跟踪输出的示例（TCP/IP 连接） (16/20)

```
17 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833156953 probe 100
 bytes 16

 Data1  (PD_TYPE_UINT,8) unsigned integer:
 10
```

图 1. 跟踪输出的示例（TCP/IP 连接） (17/20)

```
18 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833159843 probe 1177
 bytes 27

 SEND BUFFER(AR):

           RDBRLLBCK RQSDSS                  (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  000AD00100010004 200F               ........ .        ..}.......
```

图 1. 跟踪输出的示例（TCP/IP 连接） (18/20)

```
19 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943302832 probe 100
 bytes 12

 Data1  (PD_TYPE_UINT,4) unsigned integer:
 54
```

图 1. 跟踪输出的示例（TCP/IP 连接） (19/20)

```
20 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943306288 probe 1178
 bytes 71

 RECEIVE BUFFER(AR):

           ENDUOWRM RPYDSS                  (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  002BD05200010025 220C000611490004   .+.R...%"....I..  ..}.............
  0010  00162110E2E3D3C5 C3F1404040404040   ..!.......@@@@@@  ....STLEC1
  0020  4040404040400005 211502             @@@@@@..!..             .....

           SQLCARD OBJDSS                   (ASCII)           (EBCDIC)
         0 1 2 3 4 5 6 7  8 9 A B C D E F   0123456789ABCDEF  0123456789ABCDEF
  0000  000BD00300010005 2408FF             ........$..       ..}........
```

图 1. 跟踪输出的示例（TCP/IP 连接） (20/20)

相关概念:

- 第 50 页的『跟踪输出文件分析』

相关参考:

- 第 57 页的『DRDA 跟踪的后续缓冲区信息』

DRDA 跟踪的后续缓冲区信息

可以分析后续发送和接收缓冲区以获取附加信息。下一个请求包含一个落实。**commit** 命令指示主机或 iSeries 数据库服务器管理系统来落实当前工作单元。第四个缓冲区是作为落实或回滚的结果从主机或 iSeries 数据库服务器数据库管理系统中接收到的。它包含“结束工作单元应答消息”（ENDUOWRM），它指示当前工作单元已经结束。

在此示例中，跟踪条目 12 包含一个空的 SQLCA，由后接 X'FF' 的 DDM 代码点 X'2408' 指示。空的 SQLCA（X'2408FF'）指示成功（SQLCODE 0）。

第 51 页的图 1 在跟踪条目 16 中显示了包含错误 SQLCA 的接收缓冲区的示例。

相关概念:

- 第 50 页的『跟踪输出文件分析』

相关参考:

- 第 51 页的『跟踪输出文件样本』

控制中心跟踪

尝试在控制中心中跟踪问题之前，建议先确保通过从 DB2 命令提示符处输入显式命令执行操作时未发生同样的问题。通常当您在控制中心（或可从控制中心启动的一种其

他的 GUI 工具）中执行任务时，您会看到“显示命令”按钮，它将提供该工具使用的命令的确切语法。如果从 DB2 命令提示符处成功执行了这一确切命令，但在 GUI 工具中执行该命令时失败了，则应该获取控制中心跟踪。

为了获取只能在控制中心中再现的问题的跟踪，按如下所示启动控制中心:

```
db2cc -tf filename
```

这会打开控制中心跟踪并且将跟踪输出保存至指定的文件。输出文件将保存至 Windows 上的 <DB2 install path>\sqllib\tools 以及 UNIX 和 Linux 上的 /home/<userid>/sqllib/tools。

注: 启动控制中心并且启用跟踪后，使用尽可能少的步骤来再现问题。尽量避免在工具中单击不必要或不相关的项。再现问题后，关闭控制中心（以及为再现问题而打开的任何其他 GUI 工具）。

生成的跟踪文件需要发送至 DB2 支持机构以便进行分析。

相关概念:

- 『控制中心概述』（《*管理指南: 实施*》）

相关参考:

- 『db2cc - Start control center command』（*Command Reference*）

JDBC 跟踪文件

获取使用 DB2 JDBC 2 类驱动程序 Linux 版、UNIX 版和 Windows 版的应用程序的跟踪

此类型的跟踪适用于以下遇到问题的情况:

- 使用 DB2 JDBC 2 类驱动程序 Linux 版、UNIX 版和 Windows 版（DB2 JDBC 2 类驱动程序）的 JDBC 应用程序
- DB2 JDBC 存储过程

注: 有许多关键字可添加至 db2cli.ini 文件并影响应用程序行为。这些关键字可能解决应用程序问题，也可能是造成应用程序问题的原因。还有一些关键字在 CLI 文档中并未讨论。这些关键字只能从 DB2 支持机构处获取。如果 db2cli.ini 文件包含未讨论的关键字，则它们可能是 DB2 支持团队建议使用的关键字。DB2 JDBC 2 类驱动程序在内部使用 DB2 CLI 驱动程序进行数据库访问。例如，DB2 JDBC 2 类驱动程序在内部将 Java getConnection() 方法映射至DB2 CLI SQLConnect() 函数。因此，Java 开发者可能会发现 DB2 CLI 跟踪是 DB2 JDBC 跟踪的一个很好的补充。

缺省情况下，DB2 CLI/ODBC 配置关键字文件在 Windows 平台上的 sqllib 目录中和 UNIX 平台上运行 CLI/ODBC 应用程序的数据库实例的 sqllib/cfg 目录中。如果使用 ODBC 驱动程序管理器在 Windows 平台上配置用户数据源，则会在用户主（概要文件）目录中创建 db2cli.ini。还可使用环境变量 DB2CLIINIPATH 来覆盖缺省值并对该文件指定另一位置。

步骤 1. 为跟踪文件创建路径。

应该创建每个用户都有写入权限的路径。例如，在 Windows 上:

```
mkdir c:\temp\trace
```

在 Linux 和 UNIX 上:

```
mkdir /tmp/trace
chmod 777 /tmp/trace
```

步骤 2. 更新 CLI 配置关键字

可通过（A）手工编辑 db2cli.ini 文件或（B）使用 UPDATE CLI CFG 命令来执行此操作。

选项 A: 手工编辑 db2cli.ini 文件。

a. 在纯文本编辑器中打开 db2cli.ini 文件。

b. 将以下一段添加至该文件（或者如果 COMMON 段已存在，则只追加变量）:

```
[COMMON]
      JDBCTrace=1
      JDBCTracePathName=<path>
      JDBCTraceFlush=1
```

例如，其中 <path> 在 Windows 上为 C:\temp\trace，在 Linux 或 UNIX 操作系统上为 /tmp/trace。

c. 保存该文件时至少保存文件结尾的一个空白行。（这可以避免某些语法分析错误。）

选项 B: 使用 UPDATE CLI CFG 命令更新 db2cli.ini 文件。发出下列命令:

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTrace 1
      db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTracePathName <path>
```

例如，其中 <path> 在 Windows 上为 C:\temp\trace，在 Linux 或 UNIX 操作系统上为 /tmp/trace。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTraceFlush 1
```

步骤 3. 确认 db2cli.ini 配置

发出以下命令来验证是否设置并选择了正确的关键字:

```
db2 GET CLI CFG FOR SECTION COMMON
```

步骤 4. 重新启动应用程序

仅当应用程序启动时才读取 db2cli.ini 文件，因此，要使任何更改生效，都必须重新启动应用程序。

如果要跟踪 JDBC 存储过程，则表示要重新启动 DB2 实例。

步骤 5. 捕获错误

运行该应用程序直到生成错误，然后终止该应用程序。尽可能减少这种情况，在跟踪时只运行与问题再现有关的 JDBC 应用程序。这会使跟踪文件更加清晰明了。

步骤 6. 禁用 JDBC 跟踪

在 db2cli.ini 的 [COMMON] 段手工设置 JDBCTrace=0 关键字，或者发出:

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 0
      db2 UPDATE CLI CFG FOR SECTION COMMON USING JDBCTrace 0
```

重新启动正在运行和跟踪的所有应用程序。

步骤 7. 收集跟踪文件

JDBC 跟踪文件将写至 JDBCTracePathName 关键字指定的路径。生成的文件名都会以 .trc 扩展名结尾。问题再现时在跟踪路径中生成的所有文件都是必需的。

当您使用跟踪工具来诊断应用程序问题时，记住它对应用程序性能会有影响，并且会影响所有应用程序而不仅仅是测试程序。这就是一定要记住在标识问题后要关闭跟踪工具的原因。

相关概念:

- 『CLI/ODBC/JDBC trace facility』（*Call Level Interface Guide and Reference, Volume 1*）
- 『db2cli.ini initialization file』（*Call Level Interface Guide and Reference, Volume 1*）

获取使用 DB2 通用 JDBC 驱动程序的应用程序的跟踪

如果已经有正在使用 DB2 通用 JDBC 驱动程序的 SQLJ 或 JDBC 应用程序，则可使用若干不同方法来启用 JDBC 跟踪。

选择 A:

如果使用 DataSource 接口连接至数据源，则使用 DataSource.setTraceLevel() 和 DataSource.setTraceFile() 方法来启用跟踪。

选择 B:

如果使用 DriverManager 接口连接至数据源，则启用跟踪的最简单方法是获取连接之前在 DriverManager 上设置 logWriter。

例如:

```
DriverManager.setLogWriter(new PrintWriter(new FileOutputStream("trace.txt")));
```

选择 C:

如果在使用 DriverManager 接口，在装入驱动程序时还可以在 URL 中指定 traceFile 和 traceLevel 属性。例如:

```
String databaseURL = "jdbc:db2://hal:50000/sample:traceFile=c:/temp/foobar.txt;" ;
```

相关概念:

- 『Example of a trace program under the IBM DB2 Driver for JDBC and SQLJ』（*Developing Java Applications*）
- 『JDBC and SQLJ problem diagnosis with the IBM DB2 Driver for JDBC and SQLJ』（*Developing Java Applications*）

CLI 跟踪

CLI 跟踪文件

CLI 跟踪捕获有关访问 DB2 CLI 驱动程序的应用程序的信息。

CLI 跟踪基本上不提供有关 DB2 CLI 驱动程序的内部工作情况的信息。

此类型的跟踪适用于以下遇到问题的情况:

- CLI 应用程序
- ODBC 应用程序（因为 ODBC 应用程序使用 DB2 CLI 接口来访问 DB2）
- DB2 CLI 存储过程
- JDBC 应用程序和存储过程

在诊断 ODBC 应用程序时，通常最容易确定问题的方法就是使用 ODBC 跟踪或 DB2 CLI 跟踪。如果在使用 ODBC 驱动程序管理器，则可以进行 ODBC 跟踪。查阅驱动程序管理器文档以确定启用 ODBC 跟踪的方式。DB2 CLI 跟踪是特定于 DB2 的，并且通常包含的信息比一般 ODBC 跟踪要多。这两种跟踪通常极为相似，都会列示某个应用程序中所有 CLI 调用的入口点和出口点；包括这些调用的所有参数和返回码。

DB2 JDBC 2 类驱动程序 Linux 版、UNIX 版和 Windows 版（DB2 JDBC 2 类驱动程序）取决于访问数据库的 DB2 CLI 驱动程序。因此，Java 开发者可能还想启用 DB2 CLI 跟踪，以获取有关应用程序如何通过各个软件层与数据库交互的其他信息。尽管都是在 db2cli.ini 文件中设置的，但 DB2 JDBC 和 DB2 CLI 跟踪选项相互独立。

相关概念:

- 『CLI and JDBC trace files』（*Call Level Interface Guide and Reference, Volume 1*）
- 『CLI/ODBC/JDBC trace facility』（*Call Level Interface Guide and Reference, Volume 1*）

启用 CLI 跟踪

通过将特定条目添加至 db2cli.ini 文件来启用 CLI 跟踪。

注: 有许多关键字可添加至 db2cli.ini 文件并影响应用程序行为。这些关键字可能解决应用程序问题，也可能是造成应用程序问题的原因。还有一些关键字在 CLI 文档中并未讨论。这些关键字只能从 DB2 支持机构处获取。如果 db2cli.ini 文件包含未讨论的关键字，则它们可能是 DB2 支持团队建议使用的关键字。

缺省情况下，DB2 CLI/ODBC 配置关键字文件的位置在 Windows 操作系统上为 sqllib 目录，在 Linux 和 UNIX 操作系统上为运行 CLI/ODBC 应用程序的数据库实例的 sqllib/cfg 目录。如果使用 ODBC 驱动程序管理器在 Windows 上配置用户数据源，则会在用户主（概要文件）目录中创建 db2cli.ini。还可使用环境变量 DB2CLIINIPATH 来覆盖缺省值并对该文件指定另一位置。

步骤 1. 为跟踪文件创建路径。

应该创建每个用户都有写入权限的路径。例如，在 Windows 上:

```
mkdir c:\temp\trace
```

在 Linux 和 UNIX 上:

```
mkdir /tmp/trace
          chmod 777 /tmp/trace
```

步骤 2. 更新 CLI 配置关键字

可通过手工编辑 db2cli.ini 文件或使用 UPDATE CLI CFG 命令来执行此操作。

选项 A: 手工编辑 db2cli.ini 文件。

a. 在纯文本编辑器中打开 db2cli.ini 文件。

b. 将以下一段添加至该文件（或者如果 COMMON 段已存在，则只追加变量）：

```
[COMMON]
                    Trace=1
                    TracePathName=<path>
                    TraceComm=1
                    TraceFlush=1
                    TraceTimeStamp=1
                    TraceScript=1
```

例如，其中 <path> 在 Windows 上为 C:\temp\trace，在 Linux 或 UNIX 上为 /tmp/trace。

c. 保存该文件时至少保存文件结尾的一个空白行。（这可以避免某些语法分析错误。）

选项 B: 使用 UPDATE CLI CFG 命令更新 db2cli.ini 文件。发出下列命令:

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 1
   db2 UPDATE CLI CFG FOR SECTION COMMON USING TracePathName <path>
```

例如，其中 <path> 在 Windows 上为 C:\temp\trace，在 Linux 或 UNIX 上为 /tmp/trace。

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceComm 1
   db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceFlush 1
   db2 UPDATE CLI CFG FOR SECTION COMMON USING TraceTimeStamp 3
```

步骤 3. 确认 db2cli.ini 配置

发出以下命令来验证是否设置并选择了正确的关键字:

```
db2 GET CLI CFG FOR SECTION COMMON
```

步骤 4. 重新启动应用程序

仅当应用程序启动时才读取 db2cli.ini 文件，因此，要使任何更改生效，都必须重新启动应用程序。

如果要跟踪 CLI 存储过程，则表示要重新启动 DB2 实例。

步骤 5. 捕获错误

运行该应用程序直到生成错误，然后终止该应用程序。如果可以减少这种情况，在跟踪时只运行与问题再现有关的应用程序，则跟踪分析会清晰得多。

步骤 6. 禁用 CLI 跟踪

在 db2cli.ini 的 [COMMON] 段手工将 **Trace** 关键字设置为值 0，或者发出:

```
db2 UPDATE CLI CFG FOR SECTION COMMON USING Trace 0
```

然后重新启动可能正在运行和跟踪的所有应用程序。

步骤 7. 收集跟踪信息

CLI 跟踪文件将写至 **TracePathName** 关键字指定的路径。生成的文件名的格式为 <pid>t<tid>.cli，其中 <pid> 是操作系统指定的进程标识，而 <tid> 是应用程序进程生成的每个线程的数字计数器（从 0 开始）。例如，p1234t1.cli。如果要与 DB2 支持机构配合工作来诊断问题，他们会想要查看跟踪路径中生成的所有文件。

当您使用跟踪工具来诊断应用程序问题时，记住它对应用程序性能会有影响，并且会影响所有应用程序而不仅仅是测试程序。这就是一定要记住在标识问题后要关闭跟踪工具的原因。

相关概念:

- 第 61 页的『CLI 跟踪文件』
- 『CLI/ODBC/JDBC trace facility』（*Call Level Interface Guide and Reference, Volume 1*）

相关参考:

- 『CLI/ODBC configuration keywords listing by category』（*Call Level Interface Guide and Reference, Volume 1*）

解释 CLI 跟踪文件中的输入和输出参数

与任何常规函数一样，DB2 CLI 函数也有输入和输出参数。在 DB2 CLI 跟踪中，可以看到这些输入和输出参数，它们会告诉我们有关每个应用程序如何调用特定 CLI API 的大量详细信息。CLI 跟踪中显示的任何 CLI 函数的输入和输出参数都可与文档的 “CLI 参考” 部分中的 CLI 函数的定义进行比较。

以下是 CLI 跟踪文件中的某个片段:

```
SQLConnect( hDbc=0:1, szDSN="sample", cbDSN=-3, szUID="",
            cbUID=-3, szAuthStr="", cbAuthStr=-3 )
    ---> Time elapsed - +6.960000E-004 seconds

SQLRETURN   SQLConnect       (
            SQLHDBC           ConnectionHandle,  /* hdbc */
            SQLCHAR      *FAR ServerName,        /* szDSN */
            SQLSMALLINT       NameLength1,       /* cbDSN */
            SQLCHAR      *FAR UserName,          /* szUID */
            SQLSMALLINT       NameLength2,       /* cbUID */
            SQLCHAR      *FAR Authentication,    /* szAuthStr */
            SQLSMALLINT       NameLength3);      /* cbAuthStr */
```

CLI 函数的初始调用显示输入参数和指定给它们的值（如果适当的话）。

在 CLI 函数返回时，它们将显示作为结果的输出参数，例如:

```
SQLAllocStmt( phStmt=1:1 )
    <--- SQL_SUCCESS   Time elapsed - +4.444000E-003 seconds
```

在此情况下，CLI 函数 SQLAllocStmt() 将返回输出参数 phStmt，其值为 “1:1”（连接句柄 1，语句句柄 1）。

相关概念:

- 『CLI and JDBC trace files』（*Call Level Interface Guide and Reference, Volume 1*）

- 第 61 页的『CLI 跟踪文件』

相关参考:

- 『CLI and ODBC function summary』（*Call Level Interface Guide and Reference, Volume 2*）

分析 CLI 跟踪中的动态 SQL

DB2 CLI 跟踪还会显示如何通过声明并在 SQLPrepare() 和 SQLBindParameter() 中使用参数标记来执行动态 SQL。这使得您能够确定运行时执行的 SQL 语句。

以下跟踪条目显示 SQL 语句的编译过程（ “?” 表示参数标记）:

```
 SQLPrepare( hStmt=1:1, pszSqlStr=
   "select * from employee where empno = ?",
   cbSqlStr=-3 )
    ---> Time elapsed - +1.648000E-003 seconds
( StmtOut="select * from employee where empno = ?" )
SQLPrepare( )
  <--- SQL_SUCCESS   Time elapsed - +5.929000E-003 seconds
```

以下跟踪条目显示将参数标记绑定为最大长度为 7 的 CHAR:

```
SQLBindParameter( hStmt=1:1, iPar=1, fParamType=SQL_PARAM_INPUT,
fCType=SQL_C_CHAR, fSQLType=SQL_CHAR, cbColDef=7, ibScale=0,
 rgbValue=&00854f28, cbValueMax=7, pcbValue=&00858534 )
    ---> Time elapsed - +1.348000E-003 seconds
SQLBindParameter( )
    <--- SQL_SUCCESS   Time elapsed - +7.607000E-003 seconds
```

这就执行了动态 SQL 语句。rbgValue=″000010″ 显示应用程序在运行时用来取代参数标记的值:

```
SQLExecute( hStmt=1:1 )
    ---> Time elapsed - +1.317000E-003 seconds
( iPar=1, fCType=SQL_C_CHAR, rgbValue="000010" - X"303030303130",
 pcbValue=6, piIndicatorPtr=6 )
    sqlccsend( ulBytes - 384 )
    sqlccsend( Handle - 14437216 )
    sqlccsend( ) - rc - 0, time elapsed - +1.915000E-003
    sqlccrecv( )
    sqlccrecv( ulBytes - 1053 ) - rc - 0, time elapsed - +8.808000E-003
SQLExecute( )
    <--- SQL_SUCCESS   Time elapsed - +2.213300E-002 seconds
```

相关概念:

- 第 61 页的『CLI 跟踪文件』

解释 CLI 跟踪中的计时信息

有几种方法可用来从 DB2 CLI 跟踪中收集计时信息。在缺省情况下，CLI 跟踪会捕获应用程序中自上次对此特定线程执行 CLP API 调用后所花的时间。它包括客户机与服务器之间的网络时间以及 DB2 中所花的时间。例如:

```
SQLAllocStmt( hDbc=0:1, phStmt=&0012ee48 )
    ---> Time elapsed - +3.964187E+000 seconds
```

（此时间值指示应用程序中自上次调用 CLI API 后所花的时间）

```
SQLAllocStmt( phStmt=1:1 )
    <--- SQL_SUCCESS   Time elapsed - +4.444000E-003 seconds
```

（自函数完成后，此时间值指示 DB2 中所花的时间，包括网络时间）

捕获计时信息的另一个方法是使用 CLI 关键字 TraceTimeStamp。此关键字将对每次调用和 DB2 CLI API 调用的结果生成时间戳记。该关键字有 3 个显示选项，它们可能带有也可能未带 ISO 时间戳记和／或处理器时标。

在处理与计时有关的问题（如 CLI0125E – 函数顺序错误）时，它会非常有用。如果要在使用多线程应用程序时尝试确定最先发生的事件，它也会非常有用。

相关概念:

- 『CLI and JDBC trace files』（*Call Level Interface Guide and Reference, Volume 1*）
- 第 61 页的『CLI 跟踪文件』
- 『CLI/ODBC/JDBC trace facility』（*Call Level Interface Guide and Reference, Volume 1*）

相关参考:

- 『TraceTimestamp CLI/ODBC configuration keyword』（*Call Level Interface Guide and Reference, Volume 1*）

解释 CLI 跟踪中的未知值

DB2 CLI 函数可返回“未知值”作为 CLI 跟踪中的输入参数。如果 DB2 CLI 驱动程序要查找特定于该输入参数的内容，而应用程序提供了另一个值，则可能发生这种情况。例如，如果遵循 CLI 函数的过时定义或使用已经不推荐使用的 CLI 函数，则可能会发生这种情况。

您还可能会看到 CLI 函数调用返回“选项值已更改”或“键集解析器返回码”。这是显示消息的键集游标的结果，如游标因为某些特定原因降级为静态游标。

```
SQLExecDirect( hStmt=1:1, pszSqlStr="select * from org", cbSqlStr=-3 )
    ---> Time elapsed - +5.000000E-002 seconds
( StmtOut="select * from org" )
( COMMIT=0 )
( StmtOut=" SELECT A.TABSCHEMA, ..... )
( StmtOut=" SELECT A.TABSCHEMA, ..... )
( Keyset Parser Return Code=1100 )

SQLExecDirect( )
    <--- SQL_SUCCESS_WITH_INFO   Time elapsed - +1.06E+001 seconds
```

在上述 CLI 跟踪中，键集解析器指示返回码为 1100，这表示该表没有唯一索引或主键，因此未能创建键集游标。这些返回码未外部化，所以此时如果您想要有关返回码的含义的进一步信息，则需要与 DB2 支持机构联系。

调用 SQLError 或 SQLDiagRec 将指示游标类型已更改。那么，该应用程序就应该查询游标类型和并行性以确定更改的属性。

相关概念:

- 『CLI and JDBC trace files』（*Call Level Interface Guide and Reference, Volume 1*）
- 第 61 页的『CLI 跟踪文件』
- 『CLI/ODBC/JDBC trace facility』（*Call Level Interface Guide and Reference, Volume 1*）

解释多线程 CLI 跟踪输出

CLI 跟踪可跟踪多线程应用程序。跟踪多线程应用程序的最好方法是使用 CLI 关键字 TracePathName。这将生成跟踪文件 p<pid>t<tid>.cli，其中 <tid> 是应用程序的实际线程标识。

如果需要知道实际线程标识，可在 CLI 跟踪头中看到此信息:

```
[ Process: 3500, Thread: 728 ]
[ Date & Time:          02/17/2006 04:28:02.238015 ]
[ Product:              QDB2/NT DB2 v9.1.0.190 ]
...
```

还可使用 CLI 关键字 TraceFileName 将多线程应用程序跟踪写至一个文件。此方法将生成一个您选择的文件，但阅读此文件会很困难，这是因为在执行一个线程中的特定 API 的同时会执行另一个线程中的另一个 API，因此可能会导致复查跟踪时造成混乱。

通常建议将 TraceTimeStamp 设置为 ON 以便您可以通过查看执行特定 API 的时间来确定事件的真实顺序。对于调查一个线程在另一个线程中导致问题的情况，这会非常有用（如 CLI0125E - 函数顺序错误）。

相关概念:

- 『CLI and JDBC trace files』（*Call Level Interface Guide and Reference, Volume 1*）
- 第 61 页的『CLI 跟踪文件』
- 『CLI/ODBC/JDBC trace facility』（*Call Level Interface Guide and Reference, Volume 1*）

特定于平台的工具

诊断工具（Windows）

下列诊断工具可用于 Windows 2000 和 Windows XP 操作系统:

事件日志、性能监视器和其他管理工具

"管理工具"文件夹提供了各种诊断信息，包括访问事件日志和访问性能信息。

任务管理器

"任务管理器"显示正在 Windows 服务器上运行的所有进程以及有关内存使用情况的详细信息。使用此工具来查明哪些 DB2 进程正在运行并诊断性能问题。通过使用此工具，可以确定进程的内存使用情况、内存限制、已使用的交换文件空间以及内存泄漏。

要打开"任务管理器"，按 Ctrl + Alt + Delete，并从可用选项中单击**任务管理器**。

Dr. Watson

Dr. Watson 实用程序是在"常规故障保护"（GPF）事件中调用的。它记录可能有助于诊断问题的数据，并将此信息保存至文件。必须通过在命令行上输入 `drwatson` 以启动此实用程序。

随 DB2 提供的工具

DB2 提供了管理和开发工具以帮助您标识 DB2 问题（例如，管理通知日志）。

ODBC 和 CLI 跟踪

CLI 跟踪帮助您标识 CLI 和 ODBC 应用程序中的问题。

相关概念:

- 第 67 页的『诊断工具（Linux 和 UNIX）』
- 第 3 页的『首次故障数据捕获信息』

诊断工具（Linux 和 UNIX）

本节描述有关在 Linux 和 UNIX 平台上进行故障诊断和性能监视的一些基本命令。有关其中任何一个命令的详细信息，请在命令行上输入它并在前面加上 “man”。使用这些命令来收集和处理数据，这些数据有助于确定系统问题的原因。一旦收集了数据，就可由熟悉该问题的人检查它，或者，如果 DB2 支持机构要求，就交给 DB2 支持机构。

故障诊断命令（UNIX）

下列 AIX 系统命令对于 DB2 故障诊断很有用:

errpt errpt 命令报告系统错误，如硬件错误和网络故障。

- 要获取每个错误显示一行的概述，使用 **errpt**
- 要获取每个错误显示一页的更为详细的视图，使用 **errpt -a**
- 要获取错误号为 “1581762B” 的错误，使用 **errpt -a -j 1581762B**
- 要查明过去是否用光了调页空间，使用 **errpt | grep SYSVMM**
- 要查明是否存在令牌环卡或磁盘问题，检查 errpt 输出中的短语 “disk” 和 “tr0”

lsps **lsps -a** 命令监视并显示调页空间的使用情况。

lsattr 此命令显示各种操作系统参数。例如，使用以下命令来查明节点上的实内存量:

```
lsattr -l sys0 -E
```

xmperf

对于使用 Motif 的 AIX 系统，此命令启动用于收集和显示与系统相关的性能数据的图形监视器。对于每个节点，监视器都会在单个窗口中显示三维图，这对于高级监视有好处。但是，如果活动很低，则此监视器的输出将是有限值。

spmon

如果在 RS/6000® SP™ 系统上使用多个节点，您可能需要检查高性能交换机（HPS）是否正在所有工作站上运行。要查看所有节点的状态，从控制工作站使用下列其中一个命令:

- **spmon -d**，用于 ASCII 输出
- **spmon -g**，用于图形用户界面

或者，从节点工作站使用命令 **netstat -i** 来查看是否关闭了该交换机。如果已关闭高性能交换机，则节点名旁边会有一个星号（*）。例如:

```
css0* 65520 <Link>0.0.0.0.0.0
```

如果启动了高性能交换机，则不会出现星号。

故障诊断命令（Linux 和 UNIX）

下列系统命令用于所有 Linux 和 UNIX 系统（包括 AIX），除非另行说明。

df **df** 命令让您查看文件系统是否已满。

- 要查看所有文件系统（包括已安装的文件系统）中有多少可用空间，使用 **df**
- 要查看名称包含“dev”的所有文件系统中有多少可用空间，使用 **df | grep dev**
- 要查看 home 文件系统中有多少可用空间，使用 **df /home**
- 要查看文件系统“tmp”中有多少可用空间，使用 **df /tmp**
- 要查看机器上是否有足够的可用空间，检查下列命令的输出：**df /usr**、**df /var**、**df /tmp** 和 **df /home**

truss 此命令对于跟踪一个或多个进程中的系统调用很有用。

pstack

可用于 Solaris 2.5.1 或更新版本，**/usr/proc/bin/pstack** 命令显示堆栈回溯信息。**/usr/proc/bin** 目录包含用于调试以暂挂状态出现的进程的其他工具。

性能监视工具

下列工具可用于监视系统的性能。

vmstat

此命令对于确定某个对象是处于暂挂状态还是在长时间运行是很有用的。您可以监视在页面调进（pi）和页面调出（po）列中找到的页面调度速率。其他很重要的列是已分配虚拟存储器（avm）数量和可用虚拟存储器（fre）数量。

iostat 此命令对于监视 I/O 活动很有用。您可以使用读写速率来估计某些 SQL 操作所需的时间量（如果它们是系统上唯一的活动）。

netstat

此命令让您了解每个节点上的网络流量以及遇到的错误信息包的数目。它对于隔离网络问题是很有用的。

system file

可用于 Solaris 操作系统，`/etc/system` 文件包含内核配置限制的定义，例如系统上允许同时存在的最大用户数、每个用户的最大进程数以及有关资源的大小和数量的进程间通信（IPC）限制。这些限制非常重要，因为它们会影响 Solaris 操作系统机器上的 DB2 性能。

相关概念:

- 第 3 页的『首次故障数据捕获信息』
- 第 15 页的『系统核心文件（UNIX）』

相关参考:

- 第 66 页的『诊断工具（Windows）』

第 5 章 搜索知识库

如何有效地搜索已知问题

有许多资源可用于描述已知问题，包括 DB2 APAR、白皮书、红皮书、技术说明和手册。所以，有效地搜索这些（及其他）资源以迅速确定您遇到的问题是否已有解决方案是非常重要的。

在搜索之前，应清楚地了解问题的情况。

在清楚地了解问题的情况之后，需要编译搜索关键字列表以增加找到现有解决方案的机率。以下是一些技巧:

1. 在搜索中使用多个字词。您使用的搜索项越切合，搜索结果越准确。
2. 从特定结果开始，然后在必要时扩大搜索结果的范围。例如，如果返回的结果太少，则除去一些不太切合的搜索项，然后重试。或者，如果不确定要使用的关键字，可使用若干关键字进行广泛搜索，查看接收到的搜索结果类型，您就能够更准确地选择其他关键字。
3. 有时搜索特定短语会更有效。例如，如果输入 ″administration notification file″（加上引号），则只会得到包含内容与您的输入内容和顺序完全相同的短语。（相对于包含这三个单词的任何组合的所有文档）。
4. 使用通配符。如果遇到特定 SQL 错误，则搜索 “SQL5005<wildcard>”，其中 wildcard 是您要搜索的资源的相应通配符。这比只搜索 “SQL5005” 或 “SQL5005c ” 返回的结果要多。
5. 如果遇到实例异常结束并且生成陷阱文件的情况，则使用陷阱或核心文件的堆栈回溯中的前两个或前三个函数来搜索已知问题。如果返回的结果太多，则尝试添加关键字 “trap”、“abend” 或 “crash”。
6. 如果搜索特定于操作系统的关键字（如信号编号或错误值），则尝试针对常量名称而不是值进行搜索。例如，搜索 “EFGIG” 而不搜索错误号 27。

一般情况下，成功搜索到的搜索项通常包括:

- 描述所运行命令的词汇
- 描述症状的词汇
- 诊断中的标记

相关概念:

- 第 1 页的『问题确定简介』

问题确定资源

提供有大量故障诊断和问题确定信息，可帮助您使用 DB2 数据库产品。

DB2 文档:

© Copyright IBM Corp. 1993, 2006

在 DB2 信息中心以及构成 DB2 资料库的 PDF 书籍中可以找到故障诊断信息。可参阅 DB2 信息中心导航树（在浏览器窗口的左窗格中）的“支持和故障诊断”分支，以查看 DB2 故障诊断文档的完整列表。

DB2 技术支持 Web 站点:

如果您遇到了问题并且想要获取查找可能的原因和解决方案的帮助，请参阅 DB2 技术支持（DB2 Technical Support）Web 站点。该“技术站点”具有指向最新 DB2 出版物、技术说明、授权程序分析报告（APAR）、修订包的链接、内部 DB2 错误代码的最新列表以及其他资源。可搜索此知识库并查找问题的可能解决方案。

访问位于以下网址的 DB2 技术支持 Web 站点:
http://www.ibm.com/software/data/db2/udb/support

相关概念:

- 第 73 页的『DB2 技术信息概述』
- 『关于发行说明』（*发行说明*）

第 6 章 获取修订

应用修订包

DB2 修订包中包含对 IBM 公司在测试期间发现的问题（“授权程序分析报告”或“APAR”）所作的更新和修订以及对客户报告的问题所作的修订。每个修订包都附带提供了一个名为 APARLIST.TXT 的文档，它描述修订包中包含的修订。

每个修订包中都包含一个自述文件和一组发行说明:

- 修订包自述文件中提供了有关安装和卸载该修订包的指示信息。
- 发行说明中包含有关对产品所作的更改的信息。

在安装 DB2 产品之前，通过在 DB2 支持 Web 站点上选择修订包下载的链接，您可以访问并阅读修订包的自述文件和发行说明。DB2 支持 Web 站点位于以下网址: http://www.ibm.com/software/data/db2/udb/support.html。

修订包是不断累积的。这意味着任何给定版本的 DB2 的最新修订包都包含同一版本的 DB2 的先前修订包中的所有更新。建议始终在最新修订包级别运行 DB2 环境，以确保操作不会出现问题。

有两种类型的修订包映像:

- 每个单独的 DB2 产品的修订包。可以对已安装的产品应用此修订包，也可以在未安装 DB2的情况下使用此修订包来执行完整的产品安装。
- 通用修订包（仅适用于 Linux 或 UNIX）。在已经安装了多个 DB2 产品的情况下进行安装时，就可以使用通用修订包。

如果安装了本地语言，则还需要安装单独的本地语言修订包。仅当本地语言修订包与已安装的 DB2 产品处于同一修订包级别时，才能应用该本地语言修订包。如果您要应用通用修订包，则必须同时应用通用修订包和本地语言修订包来更新 DB2 产品。

当在一个多分区数据库系统上安装修订包时，该系统必须处于脱机状态，并且必须将参与实例的所有计算机都升级到同一修订包级别。

先决条件:

每个修订包都具有特定的先决条件。有关详细信息，请参阅该修订包附带提供的自述文件。

过程:

1. 通过从 DB2 支持 Web 站点（网址为 http://www.ibm.com/software/data/db2/udb/support.html）选择修订包下载的链接，访问并下载最新的 DB2 修订包。

相关参考:

- 『db2setup - Install DB2 command』（*Command Reference*）
- 『installFixPack - Update installed DB2 products command』（*Command Reference*）
- 『setup - Install DB2 command』（*Command Reference*）

© Copyright IBM Corp. 1993, 2006

附录 A. DB2 数据库技术信息

DB2 技术信息概述

DB2 技术信息可通过下列工具和方法获得:

- DB2 信息中心
  - 主题
  - DB2 工具的帮助
  - 样本程序
  - 教程
- DB2 书籍
  - PDF 文件（可下载）
  - PDF 文件（在 DB2 PDF CD 中）
  - 印刷版书籍
- 命令行帮助
  - 命令帮助
  - 消息帮助
- 样本程序

IBM 定期提供文档更新。如果访问 ibm.com® 的 DB2 信息中心上的联机版本，则不必安装文档更新，这是因为 IBM 会维护此版本的更新。如果已经安装了 DB2 信息中心，则建议安装文档更新。IBM 提供新信息时，文档更新允许更新从 *DB2 信息中心 CD* 安装或从 Passport Advantage 下载的信息。

注: DB2 信息中心主题的更新频率比 PDF 书籍或硬拷贝书籍的更新频率高。要获取最新信息，安装可用的文档更新，或者参阅 ibm.com 上的 DB2 信息中心。

可以在线访问 ibm.com 上的其他 DB2 技术信息，如技术说明、白皮书和 Redbooks™。访问位于以下网址的 DB2 信息管理软件库站点: http://www.ibm.com/software/data/sw-library/。

文档反馈

我们非常重视您对 DB2 文档的反馈。如果您想就如何改善 DB2 文档提出建议，请将电子邮件发送至 db2docs@ca.ibm.com。DB2 文档小组会阅读您的所有反馈，但不能直接答复您。请尽可能提供具体的示例，这样我们才能更好地了解您所关心的问题。如果您要提供有关具体主题或帮助文件的反馈，请加上标题和 URL。

请不要用以上电子邮件地址与 DB2 客户支持机构联系。如果您遇到文档不能解决的 DB2 技术问题，请与您当地的 IBM 服务中心联系以获得帮助。

相关概念:

- 『DB2 信息中心的功能部件』（*DB2 在线信息中心*）
- 『Sample files』（*样本主题*）

© Copyright IBM Corp. 1993, 2006

相关任务:

- 『Invoking command help from the command line processor』（*Command Reference*）
- 『Invoking message help from the command line processor』（*Command Reference*）
- 第 78 页的『更新安装在计算机或内部网服务器上的 DB2 信息中心』

相关参考:

- 第 74 页的『PDF 格式的 DB2 技术资料库』

PDF 格式的 DB2 技术资料库

下列各表描述 IBM 出版物中心（网址为 网址为 www.ibm.com/shop/publications/order）提供的 DB2 库。

尽管这些表标识书籍有印刷版，但可能未在您所在国家或地区提供它们。

这些书籍中的信息对于所有 DB2 用户来说都是基础知识；不管您是程序员、数据库管理员还是使用 DB2 Connect 或其他 DB2 产品的人员，都将发现此信息很有用。

表 1. DB2 技术信息

书名	书号	是否提供印刷版
《管理指南: 实施》	s151-0278	是
《管理指南: 计划》	s151-0280	是
Administrative API Reference	SC10-4231	是
Administrative SQL Routines and Views	SC10-4293	否
Call Level Interface Guide and Reference, Volume 1	SC10-4224	是
Call Level Interface Guide and Reference, Volume 2	SC10-4225	是
Command Reference	SC10-4226	否
《数据移动实用程序指南和参考》	s151-0284	是
《数据恢复和高可用性指南与参考》	s151-0281	是
Developing ADO.NET and OLE DB Applications	SC10-4230	是
Developing Embedded SQL Applications	SC10-4232	是
Developing SQL and External Routines	SC10-4373	否
Developing Java Applications	SC10-4233	是
Developing Perl and PHP Applications	SC10-4234	否
Getting Started with Database Application Development	SC10-4252	是

表 1. DB2 技术信息 (续)

书名	书号	是否提供印刷版
《Linux 和 Windows 上的 DB2 安装和管理入门》	g151-0300	是
《消息参考: 第 1 卷》	s151-0305	否
《消息参考: 第 2 卷》	s151-0306	否
《迁移指南》	G151-0481	是
《Net Search Extender 管理和用户指南》 **注**: 此文档的 HTML 不是从 HTML 文档 CD 安装的。	S151-0360	是
《性能指南》	s151-0279	是
Query Patroller Administration and User's Guide	GC10-4241	是
《DB2 客户机快速入门》	g151-0302	否
《DB2 服务器快速入门》	g151-0299	是
Spatial Extender and Geodetic Data Management Feature User's Guide and Reference	SC18-9749	是
SQL Guide	SC10-4248	是
SQL Reference, Volume 1	SC10-4249	是
SQL Reference, Volume 2	SC10-4250	是
《系统监视器指南和参考》	s151-0283	是
《故障诊断指南》	G151-0285	否
《Visual Explain 教程》	S151-0455	否
《新增内容》	s151-0307	是
XML Extender Administration and Programming	SC18-9750	是
《XML 指南》	s151-0282	是
XQuery Reference	SC18-9796	是

表 2. 特定于 DB2 Connect 的技术信息

书名	书号	是否提供印刷版
《DB2 Connect 用户指南》	s151-0304	是
《DB2 Connect 个人版快速入门》	G151-0431	是
《DB2 Connect 服务器快速入门》	g151-0303	是

表 3. WebSphere® 信息集成技术信息

书名	书号	是否提供印刷版
WebSphere Information Integration: Administration Guide for Federated Systems	SC19-1020	是

表 3. WebSphere® 信息集成技术信息 (续)

书名	书号	是否提供印刷版
WebSphere Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018	是
WebSphere Information Integration: Configuration Guide for Federated Data Sources	SC19-1034	否
WebSphere Information Integration: SQL Replication Guide and Reference	SC19-1030	是

注: DB2 发行说明提供特定于产品的发行版和修订包级别的附加信息。有关更多信息，请参阅相关链接。

相关概念:

- 第 73 页的『DB2 技术信息概述』
- 『关于发行说明』（*发行说明*）

相关任务:

- 第 76 页的『订购印刷版 DB2 书籍』

订购印刷版 DB2 书籍

如果您需要印刷版的 DB2 书籍，可以在许多（但不是所有）国家或地区在线购买。无论何时都可以从当地的 IBM 代表处订购印刷版的 DB2 书籍。注意，*DB2 PDF 文档* CD 上的某些软拷贝书籍没有印刷版。例如，*DB2 消息参考*的任何一卷都没有提供印刷版书籍。

只要支付一定费用，就可以从 IBM 获取 DB2 PDF 文档 CD，该 CD 包含许多 DB2 书籍的印刷版。根据您下订单的位置，您可能能够从 IBM 出版物中心在线订购书籍。如果在线订购在您所在国家或地区不可用，您总是可以从当地的 IBM 代表处订购印刷版 DB2 书籍。注意，并非 DB2 PDF 文档 CD 上的所有书籍都有印刷版。

注: 最新最完整的 DB2 文档保留在网址如下的 DB2 信息中心中：http://publib.boulder.ibm.com/infocenter/db2help/。

过程:

要订购印刷版的 DB2 书籍:

- 要了解您是否可从所在国家或地区在线订购印刷版的 DB2 书籍，可查看 IBM 出版物中心站点，网址为: http://www.ibm.com/shop/publications/order。 必须先选择国家、地区或语言才能访问出版物订购信息，然后再按照针对您所在位置的订购指示信息进行订购。
- 要从当地的 IBM 代表处订购印刷版的 DB2 书籍:
  - 从下列其中一个 Web 站点找到当地代表处的联系信息:
    - IBM 全球联系人目录，网址为 www.ibm.com/planetwide

- IBM 出版物 Web 站点，网址为 http://www.ibm.com/shop/publications/order。必须先选择国家、地区或语言才能访问对应您的所在地的出版物主页。在此页面中访问“关于此站点”链接。

– 请在致电时说明您想订购 DB2 出版物。

– 请向您当地的代表提供想要订购的书籍的书名和书号。

相关概念:

• 第 73 页的『DB2 技术信息概述』

相关参考:

• 第 74 页的『PDF 格式的 DB2 技术资料库』

从命令行处理器显示 SQL 状态帮助

DB2 返回可作为 SQL 语句结果的条件的 SQLSTATE 值。SQLSTATE 帮助说明 SQL 状态和 SQL 状态类代码的含义。

过程:

要调用 SQL 状态帮助，打开命令行处理器并输入:

`? sqlstate` 或 `? class code`

其中，*sqlstate* 表示有效的 5 位 SQL 状态，*class code* 表示该 SQL 状态的前 2 位。

例如，`? 08003` 显示 08003 SQL 状态的帮助，而 `? 08` 显示 08 类代码的帮助。

相关任务:

- 『Invoking command help from the command line processor』（*Command Reference*）
- 『Invoking message help from the command line processor』（*Command Reference*）

访问不同版本的 DB2 信息中心

对于 DB2 版本 9 主题，DB2 信息中心 URL 为 http://publib.boulder.ibm.com/infocenter/db2luw/v9/。

对于 DB2 版本 8 主题，请访问以下版本 8 信息中心 URL: http://publib.boulder.ibm.com/infocenter/db2luw/v8/。

相关任务:

- 『设置对 DB2 上下文帮助和文档的访问权』（*《管理指南: 实施》*）

以首选语言显示 DB2 信息中心中的主题

DB2 信息中心尝试以您在浏览器首选项中指定的语言显示主题。如果未提供主题的首选语言翻译版本，则 DB2 信息中心将显示该主题的英文版。

过程:

要在 Internet Explorer 浏览器中以您的首选语言显示主题:

1. 在 Internet Explorer 中，单击**工具** —> **Internet 选项** —> **语言...**按钮。“语言首选项”窗口打开。
2. 确保您的首选语言被指定为语言列表中的第一个条目。
   - 要将新语言添加至列表，单击**添加...** 按钮。

     注: 添加语言并不能保证计算机具有以首选语言显示主题所需的字体。
   - 要将语言移至列表顶部，选择该语言并单击**上移**按钮直到该语言成为语言列表中的第一个条目。
3. 清除浏览器高速缓存然后刷新页面以便以首选语言显示 DB2 信息中心。

要在 Firefox 或 Mozilla 浏览器中以首选语言显示主题:

1. 选择**工具** —> **选项** —> **语言**按钮。“语言”面板将显示在“首选项”窗口中。
2. 确保您的首选语言被指定为语言列表中的第一个条目。
   - 要将新语言添加至列表，单击**添加...** 按钮以从“添加语言”窗口中选择一种语言。
   - 要将语言移至列表顶部，选择该语言并单击**上移**按钮直到该语言成为语言列表中的第一个条目。
3. 清除浏览器高速缓存然后刷新页面以便以首选语言显示 DB2 信息中心。

在某些浏览器和操作系统组合上，可能还必须将操作系统的区域设置更改为您选择的语言环境和语言。

相关概念:

- 第 73 页的『DB2 技术信息概述』

更新安装在计算机或内部网服务器上的 DB2 信息中心

如果在本地安装了 DB2 信息中心，则可以下载已更新的主题。大多数主题底部的“最近一次更新日期”值指示该主题的当前级别。

要确定整个 DB2 信息中心是否存在更新，请查找“信息中心”主页上的“最近一次更新日期”值。将本地安装的主页中的值与 http://www.ibm.com/software/data/db2/udb/support/icupdate.html 上最新可下载更新的日期进行比较。如果提供了较新的可下载更新，就可以更新本地安装的信息中心。

更新在本地安装的 DB2 信息中心要求您:

1. 停止计算机上的 DB2 信息中心，然后以独立方式重新启动信息中心。如果以独立方式运行信息中心，则网络上的其他用户将无法访问信息中心，因而您可以下载和应用更新。
2. 使用更新功能来确定 IBM 是否提供了更新包。

   注: 在 CD 上也提供了更新。有关如何配置信息中心以从 CD 安装更新的详细信息，请参阅相关链接。

   如果提供了更新包，则使用更新功能来下载这些更新包。(更新功能只能用于独立方式。)
3. 停止独立信息中心，然后在计算机上重新启动 DB2 信息中心服务。

过程:

要更新安装在您的计算机或内部网服务器上的 DB2 信息中心:

1. 停止 DB2 信息中心服务。
   - 在 Windows 上，单击**开始 → 控制面板 → 管理工具 → 服务**。然后右键单击 **DB2 信息中心**服务，并选择**停止**。
   - 在 Linux 上，输入以下命令:

     ```
     /etc/init.d/db2icdv9 stop
     ```
2. 以独立方式启动信息中心。
   - 在 Windows 上:
     a. 打开命令窗口。
     b. 浏览至信息中心的安装路径。在缺省情况下，DB2 信息中心安装在 C:\Program Files\IBM\DB2 Information Center\Version 9 目录中。
     c. 使用 DB2 信息中心的标准路径运行 help_start.bat 文件:

        ```
        <DB2 Information Center dir>\doc\bin\help_start.bat
        ```
   - 在 Linux 上:
     a. 浏览至信息中心的安装路径。在缺省情况下，DB2 信息中心安装在 /opt/ibm/db2ic/V9 目录中。
     b. 使用 DB2 信息中心的标准路径运行 help_start 脚本:

        ```
        <DB2 Information Center dir>/doc/bin/help_start
        ```

   系统缺省 Web 浏览器将启动以显示独立信息中心。
3. 单击“更新”按钮（ ）。在信息中心的右边面板上，单击**查找更新**。将显示现有文档的更新列表。
4. 要启动下载进程，请检查您想要下载的选项，然后单击**安装更新**。
5. 在完成下载和安装进程后，单击**完成**。
6. 停止独立信息中心。
   - 在 Windows 上，使用 DB2 信息中心的标准路径运行 help_end.bat 文件:

     ```
     <DB2 Information Center dir>\doc\bin\help_end.bat
     ```

     注: help_end 批处理文件包含安全终止用 help_start 批处理文件启动的进程所需的命令。不要使用 Ctrl-C 或任何其他方法来终止 help_start.bat。
   - 在 Linux 上，使用 DB2 信息中心的标准路径运行 help_end 脚本:

     ```
     <DB2 Information Center dir>/doc/bin/help_end
     ```

     注: help_end 脚本包含安全终止用 help_start 脚本启动的进程所需的命令。不要使用任何其他方法来终止 help_start 脚本。
7. 重新启动 DB2 信息中心服务。
   - 在 Windows 上，单击**开始 → 控制面板 → 管理工具 → 服务**。然后右键单击 **DB2 信息中心**服务，并选择**启动**。
   - 在 Linux 上，输入以下命令:

     ```
     /etc/init.d/db2icdv9 start
     ```

更新后的 DB2 信息中心将显示新的主题和更新后的主题。

相关概念:

- 『DB2 信息中心安装选项』（*《DB2 服务器快速入门》*）

相关任务:

- 『使用 DB2 安装向导安装 DB2 信息中心（Linux）』（*《DB2 服务器快速入门》*）
- 『使用 DB2 安装向导安装 DB2 信息中心（Windows）』（*《DB2 服务器快速入门》*）

DB2 教程

DB2 教程帮助您了解 DB2 产品的各个方面。这些课程提供了逐步指示信息。

开始之前:

可从信息中心查看 XHTML 版的教程: http://publib.boulder.ibm.com/infocenter/db2help/ 。

某些课程使用了样本数据或代码。有关其特定任务的任何先决条件的描述，请参阅教程。

DB2 教程:

要查看教程，单击标题。

本机 XML 数据存储
设置 DB2 数据库以存储 XML 数据以及如何对本机 XML 数据存储执行基本操作。

《Visual Explain 教程》
使用 Visual Explain 来分析、优化和调整 SQL 语句以获取更好的性能。

相关概念:

- 『Visual Explain 概述』（*《管理指南: 实施》*）

DB2 故障诊断信息

有大量故障诊断和问题确定信息可帮助您使用 DB2 产品。

DB2 文档
故障诊断信息可在 DB2 信息中心的 “DB2 故障诊断指南” 或 “支持和故障诊断” 部分找到。可在该处找到有关如何使用 DB2 诊断工具和实用程序隔离和找出问题的信息、某些最常见问题的解决方案以及有关如何解决使用 DB2 产品时可能遇到的问题的建议。

DB2 技术支持 Web 站点
如果您遇到了问题并且想要获取查找可能的原因和解决方案的帮助，请参阅 DB2 技术支持 Web 站点。该 “技术支持” 站点具有指向最新 DB2 出版物、技术说明、授权程序分析报告（APAR 或错误修订）、修订包和其他资源的链接。可搜索此知识库并查找问题的可能解决方案。

访问位于以下网址的 DB2 技术支持 Web 站点: http://www.ibm.com/software/data/db2/udb/support.html

相关概念:

- 第 1 页的『问题确定简介』

• 第 73 页的『DB2 技术信息概述』

条款和条件

如果符合以下条款和条件，则授予您使用这些出版物的准用权。

个人使用：只要保留所有的专有权声明，您就可以为个人、非商业使用复制这些出版物。未经 IBM 明确同意，您不可以分发、展示或制作这些出版物或其中任何部分的演绎作品。

商业使用：只要保留所有的专有权声明，您就可以仅在企业内复制、分发和展示这些出版物。未经 IBM 明确同意，您不可以制作这些出版物的演绎作品，或者在您的企业外部复制、分发或展示这些出版物或其中的任何部分。

除非本准用权中有明确授权，不得把其他准用权、许可或权利（无论是明示的还是暗含的）授予其中包含的出版物或任何信息、数据、软件或其他知识产权。

当使用这些出版物损害了 IBM 的利益，或者根据 IBM 的规定，未正确遵守上述指导说明时，则 IBM 保留自主决定撤销本文授予的准用权的权利。

您不可以下载、出口或再出口本信息，除非完全遵守所有适用的法律和法规，包括所有美国出口法律和法规。

IBM 对这些出版物的内容不作任何保证。这些出版物“按现状”提供，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的关于适销和适用于某种特定用途的保证。

附录 B. 声明

IBM 可能不在所有国家或地区提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区: International Business Machines Corporation “按现状” 提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和／或程序进行改进和／或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的: （i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario

© Copyright IBM Corp. 1993, 2006

L6G 1C7
CANADA

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息可能包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，与实际商业企业所用的名称和地址的任何雷同纯属巧合。

版权许可:

本信息可能包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明:

©（*贵公司的名称*）（*年*）。此部分代码是根据 IBM 公司的样本程序衍生出来的。© Copyright IBM Corp. （*输入年份*）. All rights reserved.

商标

DB2 版本 9 文档库的各个文档中标识的公司、产品或服务名称可能是 International Business Machines Corporation 或其他公司的商标或服务标记。有关 IBM Corporation 在美国和／或其他国家的商标的信息在 http://www.ibm.com/legal/copytrade.shtml 中。

下列各项是其他公司的商标或注册商标，且已在 DB2 文档库中的至少一份文档中使用:

Microsoft®、Windows、Windows NT® 和 Windows 徽标是 Microsoft Corporation 在美国和／或其他国家或地区的商标。

Intel®、Itanium®、Pentium® 和 Xeon® 是 Intel Corporation 在美国和 / 或其他国家或地区的商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

索引

[A]

安装
  手工
    列出产品 32

[B]

帮助
  显示 77
  有关 SQL 语句 77

[C]

参数
  PRDID 50
错误
  问题确定 23

[D]

订购 DB2 书籍 76

[F]

发送缓冲区，跟踪数据 48
分布式数据管理（DDM） 48

[G]

跟踪
  概述 44
  输出文件 48, 49
  输出文件样本 51
  有关 DRDA 跟踪的缓冲区信息 57
  CLI 61
    分析 63, 64, 65, 66
  DB2 connect 与服务器之间的数据 48
  DRDA
    解释 48
跟踪工具
  故障诊断概述 44
  控制中心跟踪 57
  CLI 应用程序 61
  DB2 跟踪 45, 46, 47
  DRDA 跟踪 51, 57
  JDBC 应用程序 60
    跟踪选项配置 58
更新
  信息中心 78
更新 *(续)*
  DB2 信息中心 78
工具
  诊断 25, 67
    Windows 66
故障诊断 1, 17
  跟踪设施 44, 45
    控制中心跟踪 57
    CLI 和 ODBC 应用程序 61
    DRDA 51, 57
    JDBC 应用程序 58, 60
  教程 80
  联机信息 80
  连接 24
  收集信息 23, 29, 34, 42
  搜索解决方案是否存在问题 69
  问题再现 29
  资源 69
管理日志文件 5
管理通知日志 3
  解释 5

[H]

核心文件
  标识 15
  问题确定 25
  UNIX 系统 15

[J]

教程
  故障诊断和问题确定 80
  Visual Explain 80
交换服务器属性命令 50
接收缓冲区 48
结束工作单元应答消息
（ENDUOWRM） 50
进程状态实用程序 25, 50

[K]

控制中心
  跟踪 57

[L]

落实命令
  跟踪输出缓冲区 50

[M]

命令
  落实 50
  ACCRDB 50
  ACCRDBRM 50
  ACCSEC 50
  EXCSAT 50
  EXCSATRD 50
  SECCHK 50

[Q]

全局注册表
  改变 28

[R]

日志文件
  管理 5

[S]

实用程序
  跟踪 48
  进程状态 50
  db2drdat 48
  ps（进程状态） 25, 50
首次故障服务日志 25
首次故障数据捕获（FFDC）
  描述 3
  特定于平台的 14
  陷阱文件 13
  转储文件 12
数据库
  名称
    RDBNAM 对象 50
数据库分区服务器
  发出命令 21
数据库分析和报告工具命令
  故障诊断概述 27
搜索
  技巧 69

[T]

条款和条件
  出版物的使用 81
通知级别配置参数
  更新 5

© Copyright IBM Corp. 1993, 2006

[W]

文档 73, 74
  使用条款和条件 81
问题确定
  概述 1, 23
  教程 80
  联机信息 80
  连接后问题 24
  连接问题 24
  收集信息 23
  诊断工具 25
  资源 69

[X]

系统核心文件, UNIX 15
系统命令（UNIX）
  dbx 15
陷阱文件 13
  格式化（Windows） 13
信息中心
  版本 77
  更新 78
  以各种语言查看 77
修订包
  应用 71

[Y]

印刷版书籍
  订购 76
与 IBM 联系 89

[Z]

诊断工具 67
  问题确定 25
  UNIX 15
  Windows 16, 66
    访问 Dr. Watson 日志 17
诊断日志 3
诊断文件 14
注意 83
转储文件
  错误报告 12

A

ACCRDB 命令 50
ACCRDBRM 命令 50
ACCSEC 命令 50

C

CLI（调用级接口）
  跟踪
    故障诊断概述 61
  跟踪工具
    启动 61
CLI 应用程序
  跟踪工具配置 61
CLI/ODBC/JDBC
  跟踪
    故障诊断概述 61

D

DB2
  安装
    应用修订包 71
DB2 产品
  除去
    列出产品 32
  手工安装
    列出产品 32
DB2 跟踪工具（db2trc）
  转储跟踪输出 46
DB2 信息中心
  版本 77
  更新 78
  以各种语言查看 77
DB2 JDBC 驱动程序
  跟踪工具配置 60
DB2 JDBC 2 类驱动程序
  跟踪工具配置 58
db2cli.ini 文件
  跟踪配置
    CLI 和 ODBC 应用程序 61
db2cos
  输出文件 10
db2dart 命令
  故障诊断概述 27
db2diag 命令
  示例 27
db2diag.log 5
  分析 7, 8, 27
  用途 3
db2drdat 实用程序
  输出文件 48
db2level 命令
  用于故障诊断 29
db2look 命令
  用于故障诊断 29
db2pd 命令
  故障诊断示例 34
  缺省 db2cos 脚本收集的输出 10
db2support 命令
  故障诊断用法 17, 42
db2trc（DB2 跟踪工具）
  格式化跟踪输出 47
  启动, 概述 45
db2_all 命令 21
ddcstrc 实用程序
  输出文件 49
diaglevel 配置参数
  更新 7
DSS（分布式小节）
  类型, 跟踪 48

E

EXCSAT 命令 50
EXCSATRD 命令 50
EXTNAM 对象 50

F

FFDC（首次故障数据捕获） 3
  输出文件 3
  陷阱文件 13

J

JDBC 应用程序
  跟踪工具配置 58, 60

O

ODBC 应用程序
  跟踪工具配置 61

P

PRDID 参数 50
ps（进程状态）实用程序 25, 50

R

rah 命令
  简介 21

S

SECCHK 命令 50
SQL 语句
  显示帮助 77
SQLCA（SQL 通信区）
  数据的缓冲区 48
  SQLCODE 字段 48
SQLCODE
  SQLCA 中的字段 48
SRVNAM 对象 50

T

TCP/IP
  ACCSEC 命令 50
  SECCHK 命令 50
Trace 实用程序 48

U

UNIX
  除去
    列出产品 32

V

VALIDATE RUN 参数值 50
Visual Explain
  教程 80

与 IBM 联系

要与您所在国家或地区的 IBM 联系，请查看网址如下的 IBM 全球联系人目录: http://www.ibm.com/planetwide

要了解有关 DB2 产品的更多信息，请访问 http://www.ibm.com/software/data/db2/。

© Copyright IBM Corp. 1993, 2006

中国印刷

G151-0285-00

Spine information:

IBM DB2 DB2 版本 9 故障诊断指南